Patent Office
Canberra

I, JULIE BILLINGSLEY, TEAM LEADER EXAMINATION SUPPORT AND SALES hereby certify that annexed is a true copy of the Provisional specification in connection with Application No. 2002951574 for a patent by UNISEARCH LIMITED as filed on 20 September 2002.

WITNESS my hand this
Third day of October 2003

J. Billingsley

JULIE BILLINGSLEY
TEAM LEADER EXAMINATION
SUPPORT AND SALES

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION

Applicant(s):

UNISEARCH LIMITED

Invention Title:

METHOD OF SIGNALLING MOTION INFORMATION FOR EFFICIENT SCALABLE VIDEO COMPRESSION

The invention is described in the following statement:

Method of Signalling Motion Information for Efficient Scalable Video Compression

1 Field of the Invention

The present invention relaties to efficient compression of motion video sequences and, more particularly, to a method for producing a fully scalable compressed representation of the original video sequence while exploiting motion and other spatio-temporal redundancies in the source material. The invention relates specifically to the representation and signalling of motion information within a scalable compression framework which employs motion adaptive wavelet lifting steps. Additionally, the present invention relates to the estimation of motion parameters for scalable video compression and to the successive refinement of motion information by temporal resolution, spatial resolution or precision of the parameters.

2 Background of the Invention

For the purpose of the present discussion, the term "internet" will be used both in its familiar sense and also in its generic sense to identify a network connection over any electronic communications medium or collection of cooperating communications systems.

Currently, most video content which is available over the internet must be pre-loaded in a process which can take many minutes over typical modem connections, after which the video quality and duration can still be quite disappointing. In some contexts video streaming is possible, where the video is decompressed and rendered in real-time as it is being received; however, this is limited to compressed bit-rates which are lower than the capacity of the relevant network connections. The most obvious way of addressing these problems would be to compress and store the video content at a variety of different bit-rates, so that individual clients could choose to browse the material at the bit-rate and attendant quality most appropriate to their needs and patience. Approaches of this type, however, do not represent effective solutions to the video browsing problem. To see this, suppose that the video

is compressed at bit-rates of R, 2R, 3R, 4R and 5R. Then storage must be found on the video server for all these separate compressed bit-streams, which is clearly wasteful. More importantly, if the quality associated with a low bit-rate version of the video is found to be insufficient, a complete new version must be downloaded at a higher bit-rate; this new bit-stream must take longer to download, which generally rules out any possibility of video streaming.

To enable real solutions to the remote video browsing problem, scalable compression techniques are essential. Scalable compression refers to the generation of a bit-stream which contains embedded subsets, each of which represents an efficient compression of the original video with successively higher quality. Returning to the simple example above, a scalable compressed video bit-stream might contain embedded sub-sets with the bit-rates of R, 2R, 3R, 4R and 5R, with comparable quality to non-scalable bit-streams, having the same bit-rates. Because these subsets are all embedded within one another, however, the storage required on the video server is identical to that of the highest available bit-rate. More importantly, if the quality associated with a low bit-rate version of the video is found to be insufficient, only the incremental contribution required to achieve the next higher level of quality must be retrieved from the server. In a particular application, a version at rate R might be streamed directly to the client in real-time; if the quality is insufficient, the next rate-R increment could be streamed to the client and added to the previous, cached bit-stream to recover a higher quality rendition in real time. This process could continue indefinitely without sacrificing the ability to display the incrementally improving video content in real time as it is being received from the server.

The above application could be extended in a number of exciting ways. Firstly, if the scalable bit-stream also contains distinct subsets corresponding to different intervals in time, then a client could interactively choose to refine the quality associated with specific time segments which are of the greatest interest. Secondly, if the scalable bit-stream also contains distinct subsets corresponding to different spatial regions, then clients could interactively choose to refine the quality associated with specific spatial regions over specific periods of time, according to their level of interest. In a training video, for example, a remote client could interactively "revisit" certain segments of the video and continue to stream higher quality information for these segments from the server, without incurring any delay.

To satisfy the needs of applications such as that mentioned above, low bit-rate subsets of the video must be visually intelligible. In practice, this means that most of the available bits will be devoted to a low bit-rate portion of the video are likely to contribute to the reconstruction of the video at a reduced frame rate, since attempting to recover the full frame rate video over a low bit-rate channel will result in unacceptable deterioration of the spatial details within each frame. In order to achieve smooth quality scalability

within a compressed video sequence which also offers frame rate scalability, the details required to recover higher frame rates must contribute to the refinement of a model which involves motion sensitive temporal interpolation. Without temporal interpolation, missing frames cannot be introduced into a low rate video sequence without first augmenting their spatial fidelity to a level commensurate with the frames already available, and this implies a large discontinuous jump in the amount of information which must be provided to the decoder in order to smoothly increase the reconstructed video quality. Continuing this line of argument, we see that motion information is important to highly scalable video compression; moreover, the motion itself must be represented in a manner which can be scaled, according to the temporal resolution (frame rate), spatial resolution and quality of the sample data.

2.1 Motion Adaptive Transforms based on Wavelet Lifting

The present invention is best appreciated in the context of an earlier invention by Taubman, which is the subject of a patent application entitled "Method and Apparatus for Scalable Compression of Video." This earlier patent application describes a method for modifying the individual lifting steps in a lifting implementation of a temporal wavelet decomposition, so as to compensate for the effects of motion. The invention has the following advantageous properties: 1) the motion sensitive transform may be perfectly inverted, in the absence of any compression artefacts; 2) the low temporal resolution subsets of the wavelet hierarchy offer high spatial fidelity so that the transform allows excellent frame rate scalability; 3) the high pass temporal detail subbands produced by the transform have very low energy, allowing high compression efficiency; 4) in the absence of motion, the transform reduces to a regular wavelet decomposition along the temporal axis; and 5) in the presence of locally translational motion, the transform is equivalent to applying a regular wavelet decomposition along the motion trajectories.

To assist in the present discussion, we briefly summarize the key ideas behind this earlier invention. Any two-channel FIR subband transform can be described as a finite sequence of lifting steps [1]. It is instructive to begin with an example based upon the Haar wavelet transform. Up to a scale factor, this transform may be realized in the temporal domain, through a sequence of two lifting steps, as

$$h_k[\mathbf{n}] = x_{2k+1}[\mathbf{n}] - x_{2k}[\mathbf{n}]$$

$$l_k[\mathbf{n}] = x_{2k}[\mathbf{n}] + \frac{1}{2} h_k[\mathbf{n}]$$

where $x_k[\mathbf{n}] \equiv x_k[n_1, n_2]$ denotes the samples of frame k from the original video sequence and $h_k[\mathbf{n}] \equiv h_k[n_1, n_2]$ and $l_k[\mathbf{n}] \equiv l_k[n_1, n_2]$ denote the high-pass and low-pass subband frames.

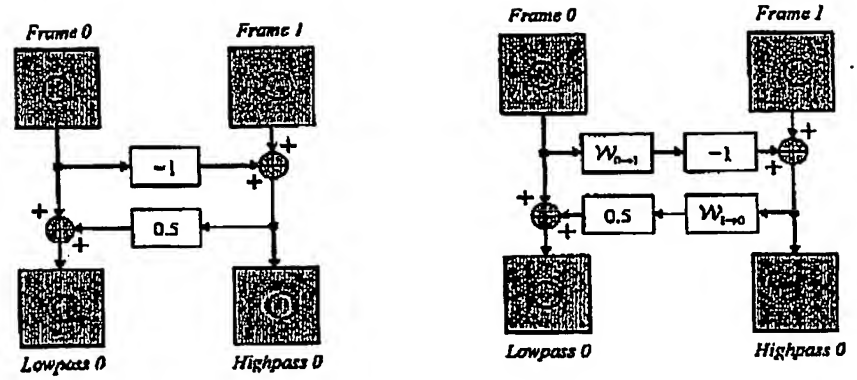


Figure 1: *Lifting representation for the Haar temporal transform (left), and its motion adaptive modification (right).*

$l_k[\mathbf{n}]$ and $h_k[\mathbf{n}]$ correspond to the scaled sum and the difference of each original pair of frames. An example is shown in Fig. 1 (left). Since motion is ignored, ghosting artefacts are clearly visible in the low-pass temporal subband, and the high-pass subband frame has substantial energy.

Now let $\mathcal{W}_{k_1 \to k_2}$ denote a motion-compensated mapping of frame k_1 onto the coordinate system of frame k_2, so that $\mathcal{W}_{k_1 \to k_2}(x_{k_1})[\mathbf{n}] \approx x_{k_2}[\mathbf{n}]$, for all $\mathbf{n}$. The lifting steps are modified as follows.

$$h_k[\mathbf{n}] = x_{2k+1}[\mathbf{n}] - \mathcal{W}_{2k \to 2k+1}(x_{2k})[\mathbf{n}] \quad (1)$$

$$l_k[\mathbf{n}] = x_{2k}[\mathbf{n}] + \frac{1}{2}\mathcal{W}_{2k+1 \to 2k}(h_k)[\mathbf{n}] \quad (2)$$

Note that $\mathcal{W}_{2k \to 2k+1}$ and $\mathcal{W}_{2k+1 \to 2k}$ represent forward and backward motion mappings, respectively. The high-pass subband frames correspond to motion-compensated residuals. These will be close to zero in regions where the motion is accurately modelled. The result is shown on the right in Fig. 1.

The framework described above is readily extended to any two-channel FIR subband transform, by motion-compensating the relevant lifting steps. We demonstrate this in the important case of the biorthogonal 5/3 wavelet transform [2]. As before, $x_{2k}[\mathbf{n}]$ and $x_{2k+1}[\mathbf{n}]$ denote the even and odd indexed frames from the original sequence. Without motion, the 5/3 transform may be implemented by alternatively updating each of these two frame sub-sequences, based on filtered versions of the other sub-sequence. The lifting

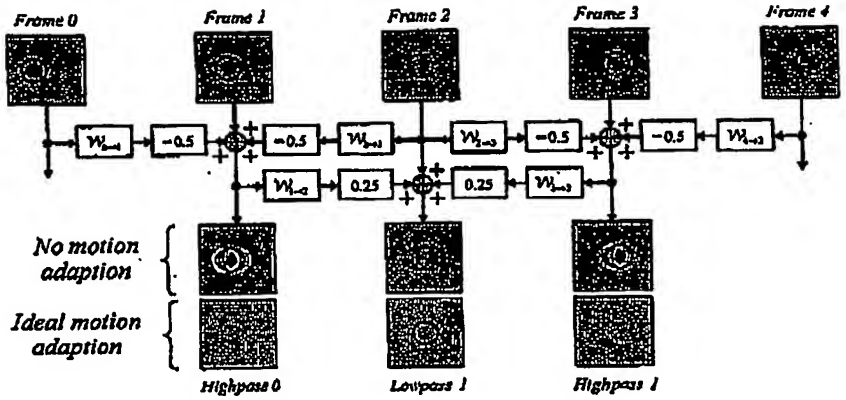


Figure 2: *Motion adaptive 5/3 temporal transform. The low-pass frame is the result of low-pass filtering along the objects motion trajectory.*

steps are

$$h_k[\mathbf{n}] = x_{2k+1}[\mathbf{n}] - \frac{1}{2}(x_{2k}[\mathbf{n}] - x_{2k+2}[\mathbf{n}])$$
$$l_k[\mathbf{n}] = x_{2k}[\mathbf{n}] + \frac{1}{4}(h_{k-1}[\mathbf{n}] + h_k[\mathbf{n}])$$

As before, we introduce motion warping operators within each lifting step, which yields the following

$$h_k[\mathbf{n}] = x_{2k+1}[\mathbf{n}] - \frac{1}{2}(\mathcal{W}_{2k\to 2k+1}(x_{2k})[\mathbf{n}] + \mathcal{W}_{2k+2\to 2k+1}(x_{2k+2})[\mathbf{n}]) \quad (3)$$
$$l_k[\mathbf{n}] = x_{2k}[\mathbf{n}] + \frac{1}{4}(\mathcal{W}_{2k-1\to 2k}(h_{k-1})[\mathbf{n}] + \mathcal{W}_{2k+1\to 2k}(h_k)[\mathbf{n}]) \quad (4)$$

Fig. 2 demonstrates the effect of these modified lifting steps. The high-pass frames are now essentially the residual from a bidirectional motion-compensated prediction of the odd-indexed original frames. When the motion is adequately captured, these high-pass frames have little energy and the low-pass frames have excellent spatial fidelity.

2.2 Counting the Cost of Motion

In the example of the Haar transform, given above, two separate motion mapping operators, $\mathcal{W}_{2k\to 2k+1}$ and $\mathcal{W}_{2k+1\to 2k}$, are required to process every pair of frames, $x_{2k}[\mathbf{n}]$ and $x_{2k+1}[\mathbf{n}]$. Their respective motion parameters must be transmitted to the decoder. To provide a larger number of temporal resolution levels, the transform is re-applied to the low-pass subband frames, $l_k[\mathbf{n}]$, for which motion mapping operators $\mathcal{W}_{4k\to 4k+2}$ and $\mathcal{W}_{4k+2\to 4k}$ are

required for every four frames. Continuing in this way, an arbitrarily large number of temporal resolutions may be obtained, using $\frac{2}{2}+\frac{2}{4}+\frac{2}{8}+\cdots \lesssim 2$ motion fields per original frame.

For the example of the 5/3 transform, also given above, four motion mapping operators, $W_{2k\rightarrow 2k+1}$, $W_{2k\rightarrow 2k-1}$, $W_{2k+1\rightarrow 2k}$ and $W_{2k-1\rightarrow 2k}$ are required for every pair of frames (indexed by k), for just one level of temporal decomposition. Continuing the transformation to an arbitrarily large number of temporal resolutions involves approximately 4 motion fields per original video frame.

The cost of estimating, coding and transmitting the above motion fields can be substantial. Moreover, this cost may adversely affect the scalability of the entire compression scheme, since it is not immediately clear how to progressively refine the motion fields without destroying the subjective properties of the reconstructed video when the motion is represented with reduced accuracy.

The previous invention clearly reveals the fact that any number of motion modeling techniques are compatible with the motion adaptive lifting transform, and also recommends the use of continuously deformable motion models such as those associated with triangular or quadrilateral meshes (see, for example, [3]). However, no particular solution is presented to the difficulties described above.

Summary of The Invention

In view of the difficulties mentioned above, a first aspect of the present invention describes a method for estimating and representing only one of the motion fields in each pair, $W_{2k\rightarrow 2k+1}$ and $W_{2k+1\rightarrow 2k}$, or $W_{2k\rightarrow 2k-1}$ and $W_{2k-1\rightarrow 2k}$. Such pairs of motion fields will be known here as "reciprocal pairs." This allows the total amount of motion information to be reduced to one motion field per frame for the Haar case, and 2 motion fields per frame for the 5/3 case. It is found that collapsing reciprocal pairs to a single motion field, from which the pair is recovered, actually improves the properties of the motion adaptive transform, resulting in increased compression efficiency, even when the benefits of reduced motion cost are not taken into account.

A second aspect of the present invention describes a method for deriving all of the required motion fields from an initial set of frame-to-frame motion fields. Thus, the compressor need only estimate the motion between each successive pair of frames, x_k [n] and x_{k+1} [n]. This substantially reduces the cost in memory and computation of the motion estimation task, without significantly altering the compression efficiency or other properties of the motion adaptive transform.

A third aspect of the invention describes a method for further reducing the motion information to 1 motion field per video frame, even for the

Note that [redacted] term "signalling" is used in [redacted] claims at the end of this application to indicate that the element referred to, if being produced by an encoder, is intended to be transmitted to a decoder.

5/3 transform. The method described herein has the property that the motion representation is temporally scalable. In particular, only one motion field must be made available to the decoder for each video frame which it can reconstruct, at any selected temporal resolution. This method involves judicious compositing of the forward and backward motion fields from different temporal resolution levels and is compatible with the efficient motion estimation method described in the second aspect.

A fourth aspect of the present invention describes efficient computational methods for performing the various motion field transformations required by other aspects of the invention. These methods must be replicated at both the compressor and the decompressor, if the transform is to remain strictly invertible.

A fifth aspect of the invention describes a method for sending approximate representations of the motion fields used for compression to the decompressor, in such a way as to balance the accuracy required for motion representation with the accuracy of the transformed sample values which may be recovered from the bit-stream. According to this aspect of the invention, a fully scalable video bit-stream may be progressively refined, both in regard to its quantized sample representations and in regard to its motion representation. This joint successive refinement property is a key element missing in the video compression literature.

The refinement method relies upon the structure of the motion representation presented in the fourth aspect. It also allows rate-distortion optimal algorithms to balance the contributions of motion information and sample accuracy, as it is being included into an incrementally improving (or layered) compressed representation. While rate-distortion optimization strategies for balancing motion and sample accuracy have been described in the literature, those algorithms were applicable only to static optimization of a compressed bit-stream for a single target bit-rate. The present invention allows for the rate-distortion optimized balancing of motion and sample accuracy to be extended to scalable content in which the target bit-rate cannot be known a priori.

3 Detailed Description of the Invention

3.1 1st Aspect: Reciprocal Motion Fields

A natural strategy for estimating the reciprocal motion fields, $\mathcal{W}_{2k\rightarrow 2k+1}$ and $\mathcal{W}_{2k+1\rightarrow 2k}$, would be to determine the parameters for $\mathcal{W}_{2k\rightarrow 2k+1}$ which minimize some measure (e.g., energy) of the mapping residual $x_{2k+1} - \mathcal{W}_{2k\rightarrow 2k+1}(x_{2k})$ and to separately determine the parameters for $\mathcal{W}_{2k+1\rightarrow 2k}$ which minimize some measure of its residual signal, $x_{2k} - \mathcal{W}_{2k+1\rightarrow 2k}(x_{2k+1})$. In general, such a procedure will lead to parameters for $\mathcal{W}_{2k\rightarrow 2k+1}$, which cannot be deduced from those for $\mathcal{W}_{2k+1\rightarrow 2k}$ and vice-versa, so that both

sets of parameters must be sent to the decoder.

It turns out that only one of the two motion fields must be directly estimated. The other can then be deduced by "inverting" the motion field which was actually estimated. Both the compressor and the decompressor may perform this inversion so that only one motion field must actually be transmitted.

True scene motion field cannot generally be inverted, due to the presence of occlusions and uncovered background. One would expect, therefore, to degrade the properties of the motion adaptive transform (e.g., compression performance, or quality of the low temporal resolution frames) by replacing $\mathcal{W}_{2k\rightarrow 2k+1}$ with an approximate inverse of $\mathcal{W}_{2k+1\rightarrow 2k}$ or vice-versa. It turns out, however, that the opposite is the case. Rather than degrading the transform, representing each reciprocal pair with only one motion field actually improves the compression efficiency and the quality of the low temporal resolution frames.

An explanation for the above phenomenon is given in [4], a copy of which is attached as APPENDIX A with this invention disclosure – the paper has not been published as of this writing. Briefly, the excellent properties of the motion adaptive temporal lifting transform are closely linked to the reciprocal relationship between the pairs, $\mathcal{W}_{2k\rightarrow 2k+1}$ and $\mathcal{W}_{2k+1\rightarrow 2k}$, and $\mathcal{W}_{2k\rightarrow 2k-1}$ and $\mathcal{W}_{2k-1\rightarrow 2k}$. If the frame warping operations described by each pair are truly inverses of one another, the motion adaptive transform is equivalent to a one-dimensional DWT, applied along the underlying motion trajectories. If they are not inverses of one another, this desirable characteristic is lost, no matter how well they are able to minimize motion compensated residuals.

According to the first aspect of the present invention, only one motion field from each reciprocal pair should be directly estimated and communicated to the decompressor. Unless otherwise prohibited (e.g., by the later aspects of the invention), it is mildly preferable to directly estimate and communicate the parameters of the motion field which is used in the first (predictive) lifting step. This is the lifting step described by equations (1) and (3), for the Haar and 5/3 cases, respectively.

3.1.1 Inversion of Triangular Mesh Motion Models

Where the motion is represented by a continuously deformable triangular mesh [3], the affine motion which describes the deformation of each triangle in $\mathcal{W}_{2k\rightarrow 2k+1}$ or $\mathcal{W}_{2k\rightarrow 2k-1}$ may be directly inverted to recover $\mathcal{W}_{2k+1\rightarrow 2k}$ and $\mathcal{W}_{2k-1\rightarrow 2k}$, respectively. A triangular mesh model for motion field $\mathcal{W}_{k_1\rightarrow k_2}$ involves a collection of node positions, $\{t_i\}$ in the target frame, $x_{k_2}[n]$, together with the locations, $\{s_i\}$ of those same node positions, as they appear in the source frame, $x_{k_1}[n]$. Although scene adaptive meshes have been described, in the preferred embodiment of the invention the target node positions, $\{t_i\}$, are fixed, and the motion field is parametrized by the set

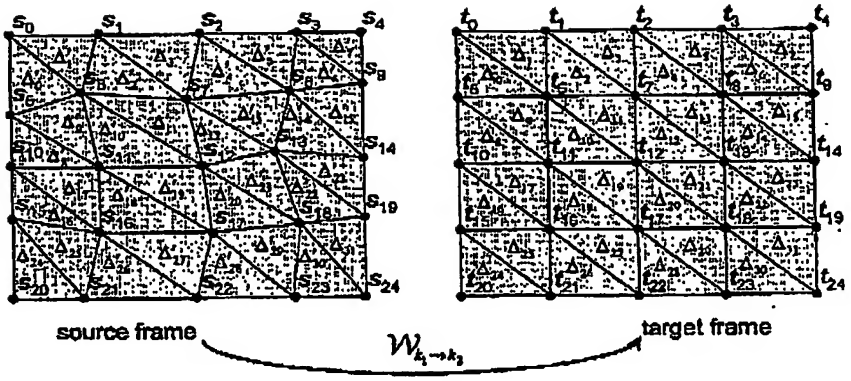


Figure 3: *Triangular mesh motion model.*

of node displacements, $\{\mathbf{s}_i - \mathbf{t}_i\}$. The target frame, $x_{k_2}[\mathbf{n}]$, is partitioned into a collection of disjoint triangles, whose vertices correspond to the node positions. Since the partition must cover the target frame, some of the target node positions must lie on the boundaries of the frame. An example involving a rectangular grid of target node vertices is shown in Fig. 3.

As suggested by the figure, it is convenient to write $\{\Delta_j\}$ for the set of target frame triangles. Let $\mathbf{t}_{j,0}$, $\mathbf{t}_{j,1}$ and $\mathbf{t}_{j,2}$ denote the vertices of target triangle Δ_j. The triangular mesh then maps the source triangle, Δ'_j, described by the vertices $\mathbf{s}_{j,0}$, $\mathbf{s}_{j,1}$ and $\mathbf{s}_{j,2}$ onto target triangle Δ_j. The motion map itself is described by an affine transformation. Specifically, for each location, $\mathbf{t} \in \Delta_j$, within the target frame, the corresponding location, $\mathbf{s}$, within the source frame is given by the affine equation

$$\mathbf{s} = A_j \mathbf{t} + \mathbf{b}_j$$

where $\mathbf{t}$, $\mathbf{s}$ and $\mathbf{b}_j$ are regarded as column vectors, A_j is a 2×2 matrix; A_j and $\mathbf{b}_j$ may be deduced from the motion parameters, using the fact that $\mathbf{t}_{j,i}$ must map to $\mathbf{s}_{j,i}$ for each $i = 0, 1, 2$. Of course, $\mathbf{s}$ does not generally lie on an integer grid, and so the source frame must be interpolated, using any of a number of well-known methods, to recover the value of $(\mathcal{W}_{k_1 \to k_2}(x_{k_1}))[\mathbf{t}]$.

In the simplest case, whenever a target node position, $\mathbf{t}_i$, lies on the boundary of frame $x_{k_2}[\mathbf{n}]$, the corresponding source node position, $\mathbf{s}_i$, is constrained to lie on the same boundary of frame $x_{k_1}[\mathbf{n}]$, as depicted in Fig. 3. In this case, the source triangles, Δ'_j, completely cover the source frame and so each location, $\mathbf{s}$, in frame $x_{k_1}[\mathbf{n}]$, may be associated with one of the triangles, Δ'_j, and hence mapped back onto the target frame through the inverse affine relation

$$\mathbf{t} = A_j^{-1}(\mathbf{s} - \mathbf{b}_j)$$

In this way, the value of $(\mathcal{W}_{k_2 \to k_1}(x_{k_2}))$ [s] may be found for each location, s, by interpolating frame x_{k_2} [n] to the location, t.

Constraining boundary nodes, t_i, to map to nodes, s_i, on the same boundary, tends to produce unrealistic motion fields in the neighbourhood of the frame boundaries, adversely affecting the ability of the mesh to track true scene motion trajectories. For this reason, the preferred embodiment of the invention does not involve any such constraints. In this case, the source triangles Δ'_j will not generally cover frame x_{k_1} [n], and inversion of the affine transformations yields values for $(\mathcal{W}_{k_2 \to k_1}(x_{k_2}))$ [s] only when s lies within one of the source triangles, Δ'_j. For locations s which do not belong to of the source triangles, Δ'_j, any of a number of policies may be described. As an example, the nearest source triangle, Δ'_j, to s may be selected and its affine parameters used to find a location t in x_{k_2} [n]. The particular policy selected for defining the inverse motion mapping at locations, s, which do not lie within any Δ'_j does not appear to have a large effect upon the performance of the motion adaptive transform, since it typically affects only those locations which are close to the frame boundaries.

3.1.2 "Inversion" of Block-Displacement Motion Models

Triangular mesh models are particularly suitable for the recovery of a reverse motion field, $\mathcal{W}_{k_2 \to k_1}$, from its forward counterpart $\mathcal{W}_{k_1 \to k_2}$. Most significantly, the transformation between target locations, t, and source locations, s, is continuous over the whole of the target frame. This is a consequence of the fact that the affine transformation maps straight lines to straight lines. Block displacement models, however, are more popular for video compression due to their relative computational simplicity. A block displacement model consists of a partition of the target frame into blocks, $\{B_i\}$, and a corresponding set of displacements, $\{\delta_i\}$, identifying the locations of each block within the source frame.

Unlike the triangular mesh, block displacement models represent the motion field in a discontinuous (piecewise constant) manner. As a result, they may not properly be inverted. Nevertheless, when reciprocal pairs of motion maps, $\mathcal{W}_{k_1 \to k_2}$ and $\mathcal{W}_{k_2 \to k_1}$, use block displacement models, it is still preferable to estimate and transmit only one of the two motion fields to the decoder, inferring the other through an approximate inverse relationship. Since displacements are usually small, it is often sufficient simply to reverse the sign of the displacement vectors, $\{\delta_i\}$, when forming $\mathcal{W}_{k_2 \to k_1}$ from $\mathcal{W}_{k_1 \to k_2}$ or vice-versa.

3.2 2nd Aspect: Compositing of Simple Motion Fields

For high energy compaction and low temporal resolution frames with high fidelity, it is essential to have accurate motion mappings for each level of a

multi-resolution temporal subband decomposition. The transform consists of a sequence of stages, each of which produces a low- and a high-pass temporal subband sequence, from its input sequence. Each stage in the temporal decomposition is applied to the low-pass subband sequence produced by the previous stage.

Since each stage of the temporal decomposition involves the same steps, one might consider applying an identical estimation strategy within each stage, estimating the relevant motion fields from the frame sequence which appears at the input to that stage. The problem with such a strategy is that estimation of the true motion, based on subband frames, may be hampered by the existence of unwanted artefacts such as ghosting. Such artefacts can arise as a result of model failure or poor motion estimation in previous stages of the decomposition.

To avoid this difficulty, it is preferred to perform motion estimation on the appropriate original frames instead of the input frames to the decomposition stage in question. For example, in the second stage of temporal decomposition it is more effective to estimate the motion mapping $\mathcal{W}^{(1)}_{k_1 \to k_2}$ between subband frames $l^{(1)}_{k_1}[\mathbf{n}]$ and $l^{(1)}_{k_2}[\mathbf{n}]$, by using the corresponding original frames $x_{k_1}[\mathbf{n}]$ and $x_{k_2}[\mathbf{n}]$. Similarly, in the third stage, it is more effective to estimate the motion mapping $\mathcal{W}^{(2)}_{k_1 \to k_2}$ between subband frames $l^{(2)}_{k_1}[\mathbf{n}]$ and $l^{(2)}_{k_2}[\mathbf{n}]$, by using the corresponding original frames $x_{4k_1}[\mathbf{n}]$ and $x_{4k_2}[\mathbf{n}]$. To clarify the notation being used here, it is noted that the first stage of decomposition employs motion mappings $\mathcal{W}^{(0)}_{k_1 \to k_2}$, producing low- and high-pass subband frames, $l^{(1)}_k[\mathbf{n}]$ and $h^{(1)}_k[\mathbf{n}]$.

After several levels of subband decomposition, the temporal displacement over which motion estimation must be performed will span many original frames. For example, in the fifth level of decomposition the actual temporal displacement between neighbouring subband frames is 16 times the original frame displacement. At a typical frame rate of 30 frames per second (fps), this corresponds to more than half a second of video.

Motion estimation is generally very difficult over large temporal displacements due to the large possible range of motion. This complexity can be reduced by using knowledge of motion mappings already obtained in previous levels of the decomposition. For example, as described by equations (3) and (4), the first stage of decomposition with the 5/3 kernel involves estimation of $\mathcal{W}^{(0)}_{2k \to 2k+1}$ and $\mathcal{W}^{(0)}_{2k+2 \to 2k+1}$. These may be composited to form an initial approximation for $\mathcal{W}^{(1)}_{k \to k+1}$, which is required for the second stage of decomposition. This is shown in Fig. 4, where the arrows indicate the direction of the motion mapping. It is often computationally simpler to create composite mappings from source mappings that have the same temporal orientation, as suggestged in the figure. If necessary, the source mappings can be inverted to achieve this. However, it is preferable to directly estimate

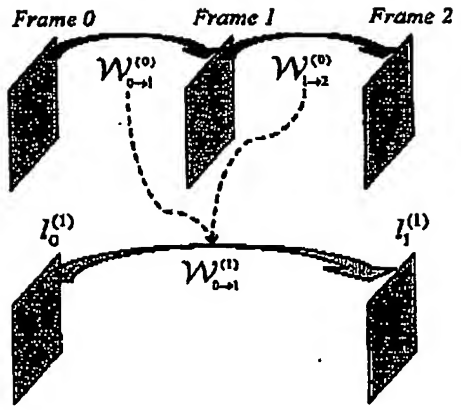


Figure 4: *Compositing of two motion fields at a higher temporal resolution to create one at a lower resolution.*

source mappings, having the same direction as the composite mapping.

The initial approximation, formed by motion field composition in the manner described above, can be refined based on original video data, using motion estimation procedures well known to those skilled in the art. It turns out, however, that the method of compositing motion fields with a frame displacement of 1 to produce motion fields corresponding to larger frame displacements often produces highly accurate motion mappings, that do not need any refinement. In some cases the composit mappings lead to superior motion adaptive transforms than motion mappings formed by direct estimation, or with the aid of refinement steps. The motion field composition method described here can be repeated throughout the temporal decomposition hierarchy so that all the mappings for the entire transform can be derived from the frame to frame motion fields estimated in the first stage.

The composition method described above eliminates a significant portion of the computational load associated with direct estimation of the required motion fields. A total of one motion mapping must be estimated for each original frame, having a temporal displacement of only one frame. This is sufficient to determine the complete set of motion mappings for the entire transform.

This method is independent of the particular wavelet kernel on which the lifting framework is based; however, the effectiveness of the composition procedure does depend on the selected motion model. An efficient method for peforming the composition procedure is described in the 4^{th} aspect of this invention.

3.3 3rd Aspect: Efficient Temporally Scalable Motion Representation

An efficient temporally scalable motion representation should satisfy two requirements. Firstly, at most one motion mapping per video frame should be needed to reconstruct the video at any temporal resolution. This is consistent with the above observation that just one mapping per frame is sufficient to derive all mappings for the entire transform.

Secondly, the above property should apply at each temporal resolution available from the transform. In particular, this means that the motion information must be temporally embedded, with each successively higher temporal resolution requiring one extra motion mapping per pair of reconstructed video frames. This property allows the video content to be reconstructed at each available temporal resolution, without recourse to redundant motion information.

This aspect of the invention involves a temporally scalable motion information hierachy, based on the method of motion field composition, as introduced in the description of the second aspect. This representation achieves both of the objectives mentioned above.

The motion information hierarchy described here is particularly important for motion adaptive lifting structures that are based on kernels longer than the simple Haar. Block transforms such as the Haar require only the motion information between every second pair of consecutive frames, at each stage of the decomposition. Therefore an efficient temporally scalable motion representation can be easily achieved by transmitting a single motion mapping for every reciprocal pair.

It is generally preferrable to use longer wavelet kernels such as the 5/3. In fact, results given in [4] reveal that this can lead to considerable improvements in performance.

The motion representation for two stages of the 5/3 transform is given in Fig. 5. The mappings required to perform the lifting steps are again shown as arrows, where the i^{th} forward mapping in the j^{th} transform level is denoted $\mathcal{F}_i^j$. The term "forward mapping" is applied to those which approximate a current frame by warping a previous frame. Likewise, backward mappings, denoted $\mathcal{B}_i^j$, correspond to warping a later frame to spatially align it with a current frame. Observe that the entire set of motion mappings depicted in Fig. 5 can be represented using only $\mathcal{F}_1^2$and $\mathcal{B}_2^1$. Inverting $\mathcal{F}_1^2$ produces the backward mapping $\mathcal{B}_1^2$. The forward mapping $\mathcal{F}_1^1$ is inferred by compositing the upper-level forward mapping $\mathcal{F}_1^2$ with the lower-level backward mapping $\mathcal{B}_2^1$. The remaining mappings $\mathcal{B}_1^1$ and $\mathcal{F}_2^1$ are recovered by inverting $\mathcal{F}_1^1$and $\mathcal{B}_2^1$, respectively.

For scenes with rapid motion, composited fields such as $\mathcal{F}_1^1$ in Fig. 5, may suffer from an accumulation of the model failure regions present in the individual mappings. If so, the compressor may correct this by transmit-

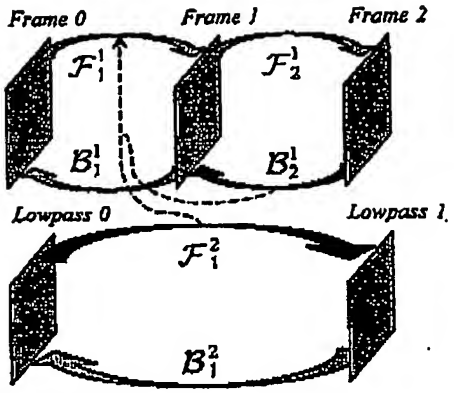


Figure 5: *Efficient temporally scalable motion representation for motion adaptive 5/3 lifting transform*

ting an optional refinement fields, possibly based on direct estimation using original data.

As mentioned, the case for the Haar wavelet is much simpler. Mappings $\mathcal{F}_2^1$and $\mathcal{B}_2^1$ are not required, so it is sufficient to code mappings $\mathcal{F}_1^1$and $\mathcal{F}_1^2$, recovering the corresponding backward motion fields by inversion.

The methods described above can be applied recursively to any number of transform stages, and the total number of required mappings is upper bounded by one per original frame. Temporal scalability is achieved since reversing a subset of the temporal decomposition stages requires no motion information from higher resolution levels.

Evidently, a motion mapping between any pair of frames can be obtained by a combination of composition and inversion operators involving the sequence of mappings $\mathcal{F}_i^2$and $\mathcal{B}_{2i}^1$. It follows that this motion representation strategy is easily modified to encompass any wavelet kernel.

3.4 4th Aspect: Efficient Implementation of Motion Field Transformations

A 4th aspect of the present invention describes an efficient method for performing the motion field composition and inversion transformations mentioned in previous aspects.

One possible way to represent a composited mapping is in terms of a sequence of warpings through each individual mapping. Motion compensation could be performed by warping the actual data through each mapping in turn. However, this approach suffers from the accumulation of spatial aliasing and other distortions that typically accompany each warping step.

A second problem with this approach is that errors due to boundary approximations also accumulate over the sequence of mappings. Boundary regions are prone to model failure, particularly when the scene undergoes global motion such as camera panning.

To avoid these problems, each location in the target frame of the composit motion field may be mapped back through the various individual mappings to find its location in the source frame of the composit motion field. The preferred method, described here, however, is to construct a triangular mesh model for the composit motion field, deducing the displacements of the mesh node points by projecting them through the various component motion mappings. The triangular mesh model provides a continuous interpolation of the projected node positions and can be represented compactly in internal memory buffers. This method is particularly advantageous when used in conjunction with triangular mesh models for all of the individual motion mappings, since the frame warping machinery required to perform the motion adaptive temporal transformation involves only one type of operation – the affine transformation described previously.

Motion field inversion may be performed using a similar strategy. The inverted motion mapping is represented using a forward triangular mesh motion model, whose node displacments are first found by tracing them through the inverse motion field. The accuracy associated with both composit and inverse motion fields representations may be adjusted by modifying the size of the triangular mesh grid. In the preferred embodiment of the invention, the mesh node spacing used for representing composit and inverse motion fields is no larger 8 frame pixels and no smaller than 4 frame pixels.

3.5 5th Aspect: Successive Refinement of Motion and Sample Accuracy

In order to provide for scalable video bit-streams which span a wide range of bit-rates, from a few 10's of kilo-bits/s (kb/s) to 10's of mega-bits/s (Mb/s), the accuracy with which motion information is represented must also be scaled. Otherwise, the cost of coding motion information would consume an undue proportion (all or more) of the overall bit budget at low bit-rates and would be insufficient to provide signficant coding gain at high bit-rates. In the 3rd aspect above, a method for providing temporally scalable motion information has been described. In this 5th aspect, a method is described for further scaling the cost of motion information, in a manner which is sensitive to both the accuracy and the spatial resolution required of the reconstructed video sequence.

During compression, an accurate motion representation is determined and used to adapt the various lifting steps in the motion adaptive transform. During decompression, however, it is not necessary to receive exactly the same motion parameters which were used during compression. The mo-

tion parameters are encoded using an embedded quantization and coding strategy. Such strategies are now well known to those skilled in the art, being employed in scalable image and video codecs such as those described in [5, 6, 7, 8]. They allow the coded bit-stream to provide a successively more accurate representation of the information being coded. For the present purposes, this information consists of the motion parameters themselves, and each motion field, $\mathcal{W}_{k_1 \to k_2}$, is provided with its own embedded bit-stream.

As an example of the way in which such an embedded motion representation may be used, consider an interactive client-server application, in which the client requests information for the video at some particular spatial resolution and temporal resolution (frame rate). Based on this information, the server determines the distortion which will be introduced by approximating the relevant motion information with only $L_q^{(M)}$ bits from the respective embedded bit-streams, where the available values for $L_q^{(M)}$ are determined by the particular embedded quantization and coding strategy which has been used. Let $D_q^{(M)}$ denote this distortion, measured in terms of Mean Squared Error (MSE), or a visually weighted MSE. The values $D_q^{(M)}$ may be estimated from the spatial-frequency power spectrum of the relevant frames. Most notably, $D_q^{(M)}$ depends not only on the accuracy with which the motion parameters are represented by the $L_q^{(M)}$ bits of embedded motion information, but also on the spatial resolution of interest. At lower spatial resolutions, less accuracy is required for the motion information, since the magnitude of the phase shifts associated with motion error are directly proportional to spatial frequency.

Continuing the example, above, the server would also estimate or know the distortion, $D_p^{(S)}$, associated with the first $L_p^{(S)}$ bits of the embedded representation generated during scalable coding of the sample values produced by the motion adaptive transform. As already noted, scalable sample data compression schemes are well known to those skilled in the art. Assuming an additive model for these two different distortion contributions, the server balances the amount of information delivered for the motion and sample data components, following the usual Lagrangian policy. Specifically, given a total budget of $L^{\max}$ bits for both components, deduced from estimates of the network transport rate, or by any other means, the server finds the largest values of p_λ and q_λ such that

$$\frac{-\Delta D_{p_\lambda}^{(S)}}{\Delta L_{p_\lambda}^{(S)}} \geq \lambda \quad \text{and} \quad \frac{-\Delta D_{q_\lambda}^{(M)}}{\Delta L_{q_\lambda}^{(M)}} \geq \lambda$$

adjusting $\lambda > 0$ so that $L_{p_\lambda}^{(S)} + L_{q_\lambda}^{(M)}$ is as large as possible, while not exceeding $L^{\max}$. Here, $\Delta D_p^{(S)}/\Delta L_p^{(S)}$ and $\Delta D_q^{(M)}/\Delta L_q^{(M)}$ are discrete approximations to the distortion-length slope at the embedded truncation points p (for sample data) and q (for motion data) respectively.

The client-server application described above is only an example. Similar techniques may be used to construct scalable compressed video files which contain an embedded hierarchy of progressively higher quality video, each level in the hierarchy having its own balance between the amount of information contributed from the embedded motion representation and the embedded sample data representation.

The strategy described above, whereby an embedded motion representation is produced by embedded quantization and coding of the individual motion parameters, may be extended to include progressive refinement according to the density of the motion parameters themselves. To see how this works, suppose that every second row and every second column were dropped from the rectangular grid of node positions in the triangular mesh of Fig. 3. In this coarse mesh, motion parameters would be sent only for the remaining node positions and the coarse triangular mesh model induced by this information would represent a coarse approximation to the original motion model. Such approximations are readily included within an embedded motion representation, from which an appropriate distribution between the motion and sample data coding costs may again be formed.

While rate-distortion optimization strategies have previously been described in the literature for balancing the costs of motion and sample data information, this has not previously been done in a scalable setting, where both the motion and the sample data accuracy are progressively refined together.

References

[1] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Applied and Computational Harmonic Analysis*, vol. 3, pp. 186–200, April 1996.

[2] D. Le Gall and A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 761–764, April 1988.

[3] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Trans. Circ. Syst. for Video Tech.*, vol. 4, pp. 339–367, Jun 1994.

[4] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *submitted to IEEE Trans. Image Proc.*, 2002.

[5] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Sig. Proc.*, vol. 41, pp. 3445–3462, December 1993.

[6] D. Taubman and A. Zakhor, "Multi-rate 3-d subband coding of video," *IEEE Trans. Image Proc.*, vol. 3, pp. 572–588, September 1994.

[7] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circ. Syst. for Video Tech.*, pp. 243–250, June 1996.

[8] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Proc.*, vol. 9, pp. 1158–1170, July 2000.

CLAIMS

1. A method for estimating and signalling motion information for a motion adaptive transform based on temporal lifting steps, said method comprising the steps of:
(a) estimating and signalling motion parameters describing a first mapping from a source frame onto a target frame within one of the lifting steps; and
(b) inferring the reciprocal mapping from said target frame onto said source frame for use within another of the lifting steps, based on the estimated and signalled motion parameters associated with said first mapping.

2. The method of Claim 1, where the motion parameters of said first mapping correspond to a deformable triangular mesh motion model.

3. The method of Claim 2, where said reciprocal mapping is inferred by inverting the affine transformations associated with the triangular mesh used to represent said first mapping.

4. The method of Claim 1, where the motion parameters of said first mapping correspond to a block displacement motion model.

5. The method of Claim 4, where said reciprocal mapping is inferred by negating the block displacement parameters associated with said first mapping.

6. The methods of any one of claims 1 through 5, where said adaptive transform involves multiple stages of temporal decomposition, corresponding to different temporal frame rates.

7. The method of Claim 6 where motion parameters at each temporal resolution are deduced from original video frames.

8. The method of claims 6 or 7, where motion parameters at lower temporal resolutions, having larger temporal displacements, are inferred by compositing the motion information estimated at higher temporal resolutions, having smaller temporal displacements.

9. The method of claims 6 or 8, where every second motion field at a higher temporal resolution is replaced by the motion field inferred by compositing the other motion field at said higher temporal resolution, that field being explicitly signalled to the
decompressor, with an appropriate motion field from the next lower temporal resolution.

10. The method of Claim 9, where said replaced motion fields are used within the lifting steps of said motion adaptive transform, in place of the originally estimated motion fields which were replaced.

11. The method of Claim 9, where said replaced motion fields are refined with additional motion parameters, said refinement parameters being signalled for use in decompression, and said replaced and refined motion fields being used within the lifting steps of said motion adaptive transform, in place of the originally estimated motion fields which were replaced.

12. The method of any one of the preceding claims, where inversion or composition of motion transformations is accomplished by applying said motion transformations to the node positions of a triangular mesh motion model, the composited or inverted motion
transformation being subsequently applied by performing the affine transformations associated with said mesh motion model.

13. A method for incrementally coding and signalling motion information for a video compression system involving a motion adaptive transform and a means for embedded coding of the transformed video samples, said method comprising the steps of:
(a) producing an embedded bit-stream, representing each motion field in coarse to fine fashion; and
(b) interleaving incremental contributions from said embedded motion fields with incremental contributions from said transformed video samples.

14. The method of Claim 13, where the embedded motion field bit-stream is obtained by applying embedded quantization and coding techniques to the motion field parameter values.

15. The method of Claim 13, where the embedded motion field bit-stream is obtained by coding the node displacement parameters associated with a triangular mesh motion model on a coarse to fine grid, each successive segment of the embedded bit-stream providing displacement parameters for node positions which lie on a finer grid than the previous stage, all coarser grids of node positions being subsets of all finer grids of node points.

16. Any of the methods mentioned in Claims 13 through 16, where said interleaving
of the contributions from the embedded motion bit-streams and from the transformed video samples is performed in a manner which minimizes the expected distortion in the reconstructed video sequence at each of a plurality of compressed video bit-rates.

17. The method of Claim 17, where the measure of distortion is Mean Squared Error.

18. The method of Claim 17, where the measure of distortion is a weighted sum of

the Mean Squared Error contributions from different spatial frequency bands, weighted according to perceptual relevance factors.

19. The method of Claim 17, where the distortion associated with inaccurate representation of the motion parameters is determined using an estimate of the spatial power spectrum of the video source.

20. The method of any one of claims 17 through 20, where the distortion associated with inaccurate representation of the motion parameters is determined using information about the spatial resolution to be used in decompressing the portion of the video bit-stream associated with the current stage of said interleaving.

21. A method of compressing a video sequence to produce a compressed video signal, including the steps of forming the video sequence into an input signal, decomposing the input signal into a set of temporal frequency bends, by separating the input signal into even and odd index frame sub-sequences, and applying a sequence of motion componsated lifting steps to alternately update the odd frame sub-sequence based upon the even frame sub-sequence and vice versa, estimating and signalling motion parameters describing a first mapping from a source frame onto a target frame within one of the lifting steps, and inferring the reciprical mapping from said target frame onto said source frame for use within another of the lifting steps, based on the estimated and signalled parameters associated with the first mapping.

22. A method for providing motion information for a motion adapted transform based on temporal lifting steps, said method comprising the steps of collapsing reciprocal pairs of motion fields to a single motion field, from which the pair of motion fields is recoverable.

23. A method of recovering motion information from the motion information provided by the method of claim 22, comprising the steps of reproducing the pair of motion fields by deducing the other motion field by inverting the single motion field provided by the method of claim 22.

24. A system for estimating and signalling motion information for a motion adaptive transform based on temporal lifting steps, said system comprising a means for estimating and signalling motion parameters describing a first mapping from a source frame onto a target frame within one of the lifting steps and means inferring the reciprocal mapping from said target frame onto said source frame for use within another of the other lifting steps, based on the estimated and signalled motion parameters associated with the said first mapping.

25. A computer program including instructions for controlling a computing system to implement the method of any one of claims 1 to 23.

26. A computer readable medium including the computer program of claim 23.

27. A system for providing motion information for a motion adapted transform based on temporal lifting steps, the system comprising means for collapsing reciprocal pairs of motion fields to a single motion field, from which the pair of motion fields is recoverable.

28. A system for recovering motion information from motion information provided by the system of claim 27, comprising means for reproducing the pair of motion fields by deducing the other motion field by inverting the single motion field provided the by system of claim 27.

APPENDIX A

Lifting-based Invertible Motion Adaptive Transform (LIMAT) Framework for Highly Scalable Video Compression

Andrew Secker and David Taubman*

Abstract

We propose a new framework for highly scalable video compression, using a Lifting-based Invertible Motion Adaptive Transform (LIMAT). We use motion-compensated lifting steps to implement the temporal wavelet transform and preserve invertibility, regardless of the motion model. By contrast, the invertibility requirement has restricted previous approaches to either block-based or global motion compensation. We show that the proposed framework effectively applies the temporal wavelet transform along the underlying motion trajectories. An implementation demonstrates high coding gain from a finely embedded, scalable compressed bit-stream. Results also demonstrate the effectiveness of wavelet kernels other than the simple Haar transform, and the benefits of complex motion modeling, using a deformable triangular mesh. Both of these advances are fundamentally incompatible with previously proposed strategies for scalable motion video compression. Video sequences reconstructed at reduced frame rates, from subsets of the compressed bit-stream, demonstrate the visually pleasing properties expected from low-pass filtering along the motion trajectories. The paper also describes a compact representation for motion parameters, having motion overhead comparable to that of motion-compensated predictive coders. While our experimental results are competitive with others reported in the literature, the principle objective of this paper is to motivate a new framework for highly scalable video compression.

1 Introduction

The objective of highly scalable video coding is to produce a dense family of embedded bit-streams, each an efficient compressed representation of the video, at successively higher bit rates. Scal-

The authors are with the School Of Electrical Engineering & Telecommunications, The University Of New South Wales, Sydney 2052, Australia. Contact: A. Secker; email: a.secker@ee.unsw.edu.au, ph: +61 (2) 9385 4803, fax: +61 (2) 9385 5993. D.Taubman; email: d.taubman@unsw.edu.au. ph: +61 (2) 9385 5223. EDICS code: 1-VIDC

able representations are important for efficient utilization of limited channel capacity, and have applications in many areas including simulcast, videoconferencing and remote video browsing. In addition to bit rate scalability, other important forms of scalability for video compression include spatial resolution and temporal (frame rate) scalability. It is desirable that all forms of scalability be achieved with little loss in performance relative to non-scalable video coders.

Highly scalable compression imposes an important restriction on the encoder. Specifically, it must operate with no prior knowledge of the rate at which the compressed video will be reconstructed. For this reason, the predictive feedback paradigm inherent in traditional motion-compensated video compression algorithms is fundamentally incompatible with highly scalable compression. Instead the preferred method is that of feed-forward compression, in which a spatio-temporal decomposition is followed by quantization and coding.

Karlsson and Vetterli first proposed the use of the separable three-dimensional (3D) Discrete Wavelet Transform (DWT) for video compression [1]. Separable 3D transforms are also employed in [2], which extends the well-known SPIHT [3] coding algorithm to the temporal dimension. However, without motion compensation, temporal filtering produces visually disturbing ghosting artefacts in the low-pass temporal subband. This is clearly undesirable where temporal scalability is of interest. The challenge therefore lies in finding a way to effectively exploit motion within the spatio-temporal transform.

Taubman and Zakhor [4] proposed an approach based on spatially aligning video frames prior to the application of the separable 3D-DWT. The spatial alignment may be achieved by arbitrary frame warping operations, but invertible warping operations cannot represent the local expansion and contraction effects exhibited within most video sequences. This is because a warping involving local expansion and contraction essentially corresponds to a non-uniform resampling of the original frame, violating the Nyquist sampling criterion in regions of expansion. In some proposed schemes, such as that of Tham et al. [5], the invertibility requirement is deliberately violated, so that high quality reconstruction is impossible.

Another class of approaches can be described as block displacement methods, originally proposed by Ohm [6] and later by Choi and Woods [7]. In these approaches, video frames are divided into blocks, where each block undergoes rigid motion, usually translation. The 3D-DWT is essentially applied in a separable fashion to the displaced blocks, but the effects of expansion and contraction in the motion field are observed in the appearance of "disconnected" pixels between the blocks. To retain invertibility, these disconnected pixels must be treated differently, seriously impairing compression performance. In addition, perfect reconstruction is impossible for non-integer block displacements, although extensions to half-pixel accuracy have been made with relatively little loss in performance [6], [7]. These approaches are also limited to block-based rigid motion models, which cannot capture expansive or contractive motion. Furthermore, the tight coupling between the temporal transform and the motion model also hampers the use of wavelet kernels other than the Haar in the temporal dimension [6], [4].

In this paper, we propose a Lifting-based Invertible Motion Adaptive Transform (LIMAT), which overcomes the limitations of the existing methods mentioned above. We extend and elaborate on preliminary work previously published in [8]. The proposed framework employs a lifting realization of the temporal DWT, in which each lifting step is compensated for the estimated scene motion. The LIMAT framework enables the construction of motion adaptive transforms with any wavelet kernel or motion model, without compromising invertibility.

Section 2 describes the proposed framework, beginning with examples based on the Haar and 5/3 wavelet kernels. By contrast to previous approaches, our results indicate that the 5/3 offers superior compression performance compared to the Haar transform. In the context of spatially continuous frames, we show that the LIMAT framework is equivalent to applying the temporal transform along the scene motion trajectories. In the discrete spatial domain, this behaviour is retained for the most important spatial frequencies, and the remaining higher frequencies are dealt with in a way that preserves the invertibility of the transform.

In practice, the performance of a motion adaptive transform is inevitably dependent on the

properties of the selected motion model. In Section 3 we provide the example of a deformable mesh motion model, using a simple block motion model as our reference. Unlike block-based motion models, deformable meshes are able to represent local expansions and contractions, while maintaining a continuous motion field. This can improve the effectiveness of motion compensation. Perhaps even more importantly, only in the context of a continuous motion field can the proposed transform be truly seen as filtering along the motion trajectories. Accordingly, our results indicate improved compression with the deformable mesh, as compared to a block-based motion model.

In Section 4, we propose an efficient representation for the motion information associated with the LIMAT framework, providing examples for the Haar and 5/3 transforms. This representation exploits the significant redundancy between the distinct motion mappings involved in the transform, so that the cost of coding the motion information is comparable with motion-compensated predictive coders.

Our experiments are described in detail in Section 5 where we provide compression results for several standard test sequences. In these experiments, the temporal transform is followed by spatial wavelet decomposition and embedded block coding, using an implementation of the JPEG2000 image compression standard. Our results are compared with those reported by another motion-compensated 3D subband coder, but note that the purpose of this paper is not to propose a complete video compression scheme. In particular, further development of the motion modelling is likely to be beneficial. Notwithstanding this, our example incarnation of the LIMAT framework achieves significantly higher compression than the reference coder.

2 Motion Adaptive Temporal DWT Based on Lifting

The key to efficient scalable video coding is to effectively exploit motion within the spatio-temporal transform. Specifically, we identify three primary objectives for a motion adaptive temporal transform, suitable for highly scalable video compression. Firstly, the low-pass temporal subband frames must represent a high quality reduced frame rate video. In particular, the transform should not

introduce ghosting artefacts into the low-pass frames, so the visual quality is comparable to that obtained by temporally subsampling the original video sequence prior to compression. In fact, if the low-pass temporal subband frames are obtained by filtering along the true scene motion trajectories, the reduced frame rate video sequence obtained by discarding high temporal frequency subbands from the compressed representation, may have an even higher quality than that obtained by subsampling the original video sequence. This is because low-pass filtering along the motion trajectories tends to reduce camera noise and scene illuminant variations.

Secondly, the transform should exhibit high coding gain. This means the high-pass temporal subband frames should contain as little energy as possible, after adjusting for the energy gains associated with the subband synthesis system. It is also important to minimize the introduction of spurious spatial details in both the high-pass and low-pass frames, since these hamper the effectiveness of subsequent spatial transformation and coding techniques. If the subbands are obtained by applying suitable wavelet filters along the true scene motion trajectories, there should be minimal introduction of spurious spatial details, and the high-pass subbands should have particularly low energy. Moreover, so long as the quality of the low-pass temporal subband frames is preserved, iterative application of the temporal decomposition along the low-pass channel should yield a multi-resolution hierarchy with similar properties.

As suggested, the above objectives are both closely related to the use of temporal filtering and subsampling along the motion trajectories associated with a realistic motion model. However, realistic motion models inevitably accommodate expansion and contraction, which makes it difficult to achieve our third objective, that the transform should be invertible. Of course, lossy compression generally prevents the original video sequence from being recovered exactly, but lack of invertibility in the transform limits the range of compressed bit rates over which the complete compression system can be used efficiently.

In the LIMAT framework, the key to accomplishing invertibility is the use of lifting [9]. Any two-channel FIR subband transform can be described as a finite sequence of lifting steps. In this

scheme, the original lifting steps are modified to exploit motion, which in no way compromises the invertibility of the transform. In fact, by introducing integer rounding into our modified lifting steps, in the manner suggested by Calderbank et al. [10], it is possible to achieve efficient lossless compression of the original video sequence. Most significantly, however, the lifting-based transform may be understood as applying the temporal wavelet transform along the underlying motion trajectories of the scene.

2.1 Example with the Haar Transform

It is instructive to begin with an example based upon the Haar wavelet transform. Up to a scale factor, this transform may be realized in the temporal domain, through a sequence of two lifting steps, as

$$h_k[\mathbf{n}] = x_{2k+1}[\mathbf{n}] - x_{2k}[\mathbf{n}]$$

$$l_k[\mathbf{n}] = x_{2k}[\mathbf{n}] + \frac{1}{2}h_k[\mathbf{n}]$$

where $x_k[\mathbf{n}] \equiv x_k[n_1, n_2]$ denotes the samples of frame k from the original video sequence and $h_k[\mathbf{n}] \equiv h_k[n_1, n_2]$ and $l_k[\mathbf{n}] \equiv l_k[n_1, n_2]$ denote the high-pass and low-pass subband frames. This decomposition of the Haar transform into two steps is also known as the S-transform [11].

The reader can verify that $l_k[\mathbf{n}]$ and $h_k[\mathbf{n}]$ correspond to the scaled sum and the difference of each original pair of frames. An example is shown in Fig. 1. Since motion is ignored, ghosting artefacts are clearly visible in the low-pass temporal subband, and the high-pass subband frame has substantial energy.

Now let $\mathcal{W}_{k_1 \to k_2}$ denote a motion-compensated mapping of frame k_1 onto the coordinate system of frame k_2, so that $\mathcal{W}_{k_1 \to k_2}(x_{k_1})[\mathbf{n}] \approx x_{k_2}[\mathbf{n}]$, for all $\mathbf{n}$. No particular motion model is assumed here. The lifting steps are modified as follows.

$$\begin{aligned} h_k[\mathbf{n}] &= x_{2k+1}[\mathbf{n}] - \mathcal{W}_{2k \to 2k+1}(x_{2k})[\mathbf{n}] \\ l_k[\mathbf{n}] &= x_{2k}[\mathbf{n}] + \frac{1}{2}\mathcal{W}_{2k+1 \to 2k}(h_k)[\mathbf{n}] \end{aligned}$$

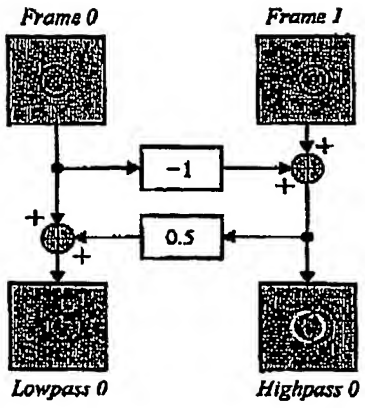


Figure 1: Lifting representation for the Haar temporal transform.

Observe that $\mathcal{W}_{2k\to 2k+1}$ and $\mathcal{W}_{2k+1\to 2k}$ represent forward and backward motion mappings, respectively. The high-pass subband frames correspond to motion-compensated residuals. These will be close to zero in regions where the motion is accurately modelled. As we shall see in Section 2.3, so long as the motion is well modelled by $\mathcal{W}_{2k\to 2k+1}$ and $\mathcal{W}_{2k+1\to 2k}$, the low-pass frames $l_k[\mathbf{n}]$, are effectively the result of applying a low-pass temporal filter along the motion trajectories. For the Haar wavelet, this low-pass analysis filter has transfer function

$$F_0(z) = \frac{1}{2}(1+z)$$

The visual effects of motion compensation can be seen by comparison of Figs. 1 and 2. In the example of Fig. 2, we assume that the motion is captured perfectly. As a result, the high-pass frame has no energy, and the low-pass frame is an excellent representation of frame 0, free from the ghosting artefacts observed in Fig. 1. If the signal amplitude on the surface of the moving object fluctuates over time, possibly due to noise, the low-pass frame represents a temporal average of the object's surface intensity, while the high-pass frame represents the temporal difference.

The modified Haar lifting steps evidently achieve the first two objectives identified above, in spatial regions where the motion is well modelled. Moreover, invertibility is an inherent property of the lifting structure, regardless of how we choose to compensate for motion. To invert the

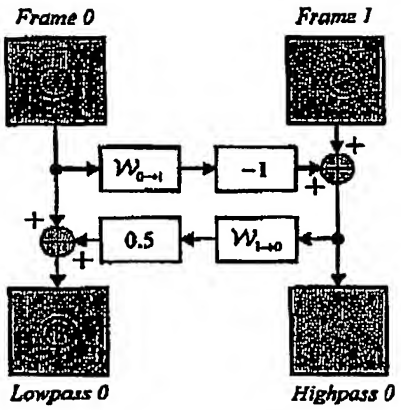


Figure 2: Same as Figure 1, but with motion-compensated lifting steps. This avoids ghosting in the low-pass frames and reduces the energy in the high-pass frames.

transform, one must simply apply the same lifting steps in reverse order, reversing the sign of the updates. The reverse transform in this example is given by

$$\begin{aligned} x_{2k}[\mathbf{n}] &= l_k[\mathbf{n}] - \frac{1}{2}\mathcal{W}_{2k+1\to 2k}(h_k)[\mathbf{n}] \\ x_{2k+1}[\mathbf{n}] &= h_k[\mathbf{n}] + \mathcal{W}_{2k\to 2k+1}(x_{2k})[\mathbf{n}]) \end{aligned}$$

2.2 A More Interesting Wavelet Transform

The framework described above is readily extended to any two-channel FIR subband transform, by motion-compensating the relevant lifting steps. We demonstrate this in the important case of the biorthogonal 5/3 wavelet transform [12], whose lifting incarnation plays an important role in the highly scalable JPEG2000 image compression standard [13]. As before, $x_{2k}[\mathbf{n}]$ and $x_{2k+1}[\mathbf{n}]$ denote the even and odd indexed frames from the original sequence. Without motion, the 5/3 transform may be implemented by alternatively updating each of these two frame sub-sequences, based on filtered versions of the other sub-sequence. The lifting steps are

$$\begin{aligned} h_k[\mathbf{n}] &= x_{2k+1}[\mathbf{n}] - \frac{1}{2}(x_{2k}[\mathbf{n}] + x_{2k+2}[\mathbf{n}]) \\ l_k[\mathbf{n}] &= x_{2k}[\mathbf{n}] + \frac{1}{4}(h_{k-1}[\mathbf{n}] + h_k[\mathbf{n}]) \end{aligned}$$

As before, we introduce arbitrary motion warping operators within each lifting step, which yields the following

$$\begin{aligned} h_k[\mathbf{n}] &= x_{2k+1}[\mathbf{n}] - \frac{1}{2}(\mathcal{W}_{2k\to 2k+1}(x_{2k})[\mathbf{n}] + \mathcal{W}_{2k+2\to 2k+1}(x_{2k+2})[\mathbf{n}]) \\ l_k[\mathbf{n}] &= x_{2k}[\mathbf{n}] + \frac{1}{4}(\mathcal{W}_{2k-1\to 2k}(h_{k-1})[\mathbf{n}] + \mathcal{W}_{2k+1\to 2k}(h_k)[\mathbf{n}]) \end{aligned}$$

In Fig. 3 we see the effect of these modified lifting steps. The high-pass frames are now essentially the residual from a bidirectional motion-compensated prediction of the odd-indexed original frames. When the motion is adequately captured, these high-pass frames have little energy and the low-pass frames correspond to low-pass filtering of the original video sequence along its motion trajectories. In this example, the surface intensity of the moving object varies randomly over time. These variations are effectively low-pass filtered, improving the visual quality of the low-pass temporal subband. The low-pass analysis filter in this case has transfer function

$$H_0(z) = -\frac{1}{8}z^2 + \frac{1}{4}z + \frac{3}{4} + \frac{1}{4}z^{-1} - \frac{1}{8}z^{-2}$$

whose frequency response is much closer to that of an ideal low-pass filter than was the Haar filter. As before, failure to capture the motion reduces the coding gain, and introduces multiple ghosting artefacts into the low-pass subband frames, as suggested by the figure.

In our experiments, the 5/3 wavelet consistently outperforms the Haar transform, which contrasts with observations reported in the context of the pre-warping and block displacement methods [14, 4]. Table 1 presents indicative results, comparing the behaviour of the Haar and 5/3 wavelet kernels. These results were obtained using block-based motion warping operators, over three levels of temporal transform. The reconstruction bit rate is 1 Mbps, but similar results were obtained at other bit rates. A full description of the experimental conditions associated with these and other results presented in this paper may be found in Section 5.

According to Table 1, the 5/3 transform provides an improvement in PSNR of between 1.15 dB and 2.64 dB, relative to the Haar transform. This can be attributed to improved low-pass filtering along motion trajectories, as well as simultaneous use of both forward and backward

Table 1: Reconstructed PSNR using 5/3 and Haar wavelets at 1 Mbps

Sequence	Haar	5/3	Gain
Mobile and Calender	26.34	27.49	+1.15
Table Tennis	30.45	33.09	+2.64
Flower Garden	26.39	28.21	+1.82
Football	25.02	27.30	+2.28

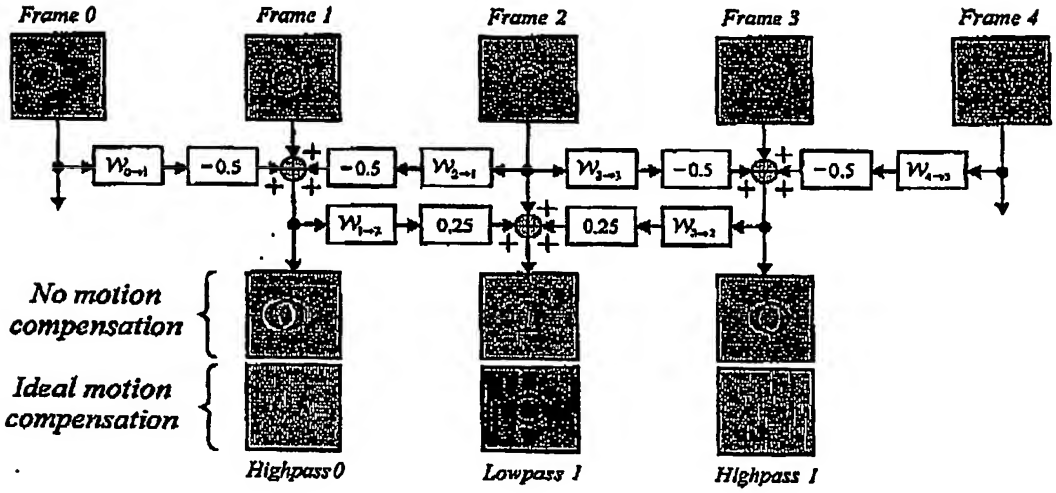


Figure 3: Motion adaptive 5/3 temporal transform. The low-pass frame is the result of low-pass filtering along the objects motion trajectory.

motion compensation to reduce the effects of inaccuracy in the motion model. Note that these results exclude the cost of coding the motion information, and the 5/3 transform has twice as many distinct motion mappings as the Haar. In Section 4 we propose a motion representation that can potentially reduce the number of coded motion mappings to one per original frame, for any transform.

2.3 Generalization and Interpretation of the Lifting Transform

We have already mentioned that motion-compensating the lifting steps of a temporal subband transform effectively results in the relevant subband filters being applied along the motion trajectories described by the motion model. In this section, we provide justification for this statement. To do so, we first consider the application of the motion-compensated lifting transform to a sequence

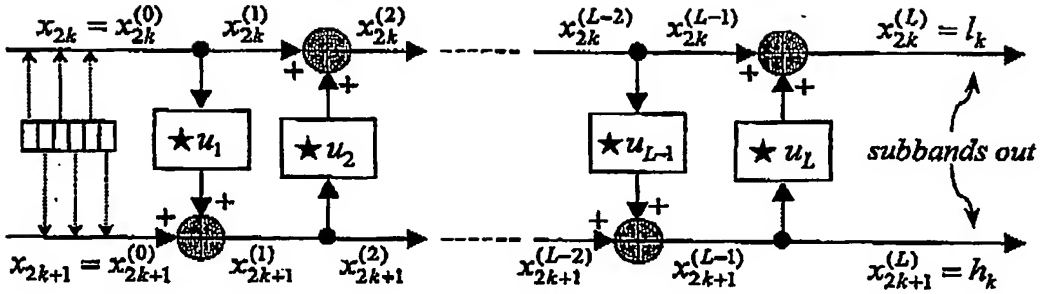


Figure 4: Network of L lifting steps, used to realize a two-channel subband transform, with subband sequences l_k and h_k.

of spatially continuous video frames, denoted $x_k(\mathbf{s}) \equiv x_k(s_1, s_2)$, where $\mathbf{s} \in \mathbb{R}^2$ represents the continuous spatial location and $k \in \mathbb{Z}$ is the frame index.

It is known that any two-channel FIR subband transform may be factored into a finite sequence of Λ lifting steps. Each successive lifting step converts its input sequence, denoted $x_k^{(\lambda-1)}$, into an output sequence, $x_k^{(\lambda)}$, where $\lambda = 1, 2, \ldots, \Lambda$, and $x_k^{(0)} \triangleq x_k$ is the input sequence supplied to the subband transform. For odd λ, the odd indexed sub-sequence, $x_{2k+1}^{(\lambda-1)}$ is updated using a filtered version of the even indexed sub-sequence, $x_{2k}^{(\lambda-1)}$, according to

$$x_{2k+1}^{(\lambda)} = x_{2k+1}^{(\lambda-1)} + \sum_i u_i^{(\lambda)} \cdot x_{2(k-i)}^{(\lambda-1)}$$

Here $u_i^{(\lambda)}$ denotes the impulse response of the λ^{th} lifting step filter. For even λ, the even sub-sequence is updated using a filtered version of the odd sub-sequence, according to

$$x_{2k}^{(\lambda)} = x_{2k}^{(\lambda-1)} + \sum_i u_i^{(\lambda)} \cdot x_{2(k-i)+1}^{(\lambda-1)}$$

The even and odd sub-sequences output from the final lifting step are the low-pass and high-pass subband sequences, respectively. That is

$$l_k = x_{2k}^{(\Lambda)} \quad \text{and} \quad h_k = x_{2k+1}^{(\Lambda)}$$

The succession of lifting steps is illustrated in Fig. 4 and its inverse is illustrated in Fig. 5.

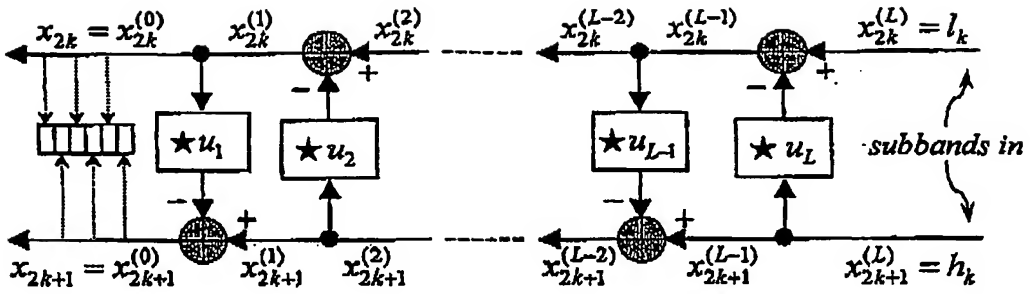


Figure 5: Synthesis network corresponding to the analysis system in Figure 4.

Using this notation, our motion-compensated temporal transform may be expressed through the lifting steps

$$x_{2k+1}^{(\lambda)}(\mathbf{s}) = x_{2k+1}^{(\lambda-1)}(\mathbf{s}) + \sum_i u_i^{(\lambda)} \cdot \mathcal{W}_{2(k-i)\to 2k+1}\left(x_{2(k-i)}^{(\lambda-1)}\right)(\mathbf{s}), \quad \lambda \text{ odd}$$

$$x_{2k}^{(\lambda)}(\mathbf{s}) = x_{2k}^{(\lambda-1)}(\mathbf{s}) + \sum_i u_i^{(\lambda)} \cdot \mathcal{W}_{2(k-i)+1\to 2k}\left(x_{2(k-i)+1}^{(\lambda-1)}\right)(\mathbf{s}), \quad \lambda \text{ even}$$

Suppose now that our motion model is invertible, meaning that there is a one to one correspondence between locations $\mathbf{s}$ in frame 0 and locations $\mathbf{s}_k = V_k(\mathbf{s})$ in frame k. Equivalently, we are assuming that our motion model assigns unique trajectories, represented by the sequence, $\{\mathbf{s}_k\}$, to each initial location in frame 0 such that the trajectories do not intersect. In this assumption, we are clearly ignoring the finite spatial support of the frames, as well as the possibility of occlusion.

Since the motion model is invertible, we must have

$$\mathcal{W}_{k_1\to k_2}(x_{k_1})(\mathbf{s}_{k_2}) = x_{k_1}\left(V_{k_1}\left(V_{k_2}^{-1}(\mathbf{s}_{k_2})\right)\right)$$

That is, to find the location $\mathbf{s}_{k_1}$ in frame k_1 which corresponds to a location $\mathbf{s}_{k_2}$ in frame k_2, we can map $\mathbf{s}_{k_2}$ back to the origin of its motion trajectory in frame 0, through $V_{k_2}^{-1}$, and then map it forward along the same trajectory into frame k_1, using V_{k_1}. We can now rewrite the motion-compensated lifting steps in terms of the sequence of locations $\mathbf{s}_k = V_k(\mathbf{s})$, corresponding to a

single motion trajectory anchored at location s in frame 0. We get

$$\begin{aligned} x_{2k+1}^{(\lambda)}(s_{2k+1}) &= x_{2k+1}^{(\lambda-1)}(s_{2k+1}) + \sum_i u_i^{(\lambda)} \cdot \mathcal{W}_{2(k-i)\to 2k+1}\left(x_{2(k-i)}^{(\lambda-1)}\right)(s_{2k+1}) \\ &= x_{2k+1}^{(\lambda-1)}(s_{2k+1}) + \sum_i u_i^{(\lambda)} \cdot x_{2(k-i)}^{(\lambda-1)}\left(s_{2(k-i)}\right), \quad \lambda \text{ odd} \\ x_{2k}^{(\lambda)}(s_{2k}) &= x_{2k}^{(\lambda-1)}(s_{2k}) + \sum_i u_i^{(\lambda)} \cdot \mathcal{W}_{2(k-i)+1\to 2k}\left(x_{2(k-i)+1}^{(\lambda-1)}\right)(s_{2k}) \\ &= x_{2k}^{(\lambda-1)}(s_{2k}) + \sum_i u_i^{(\lambda)} \cdot x_{2(k-i)+1}^{(\lambda-1)}\left(s_{2(k-i)+1}\right), \quad \lambda \text{ even} \end{aligned}$$

Finally, let $\tilde{x}_k(s)$ denote the warped frame obtained by mapping $x_k(s)$ from the coordinate system associated with frame k onto the coordinate system associated with frame 0. That is,

$$\tilde{x}_k(s) = \mathcal{W}_{k\to 0}(x_k)(s) = x_k(V_k(s))$$

The lifting steps may be expressed in terms of the sequence of warped frames as

$$\begin{aligned} \tilde{x}_{2k+1}^{(\lambda)}(s) &= \tilde{x}_{2k+1}^{(\lambda-1)}(s) + \sum_i u_i^{(\lambda)} \cdot \tilde{x}_{2(k-i)}(s), \quad \lambda \text{ odd} \\ \tilde{x}_{2k}^{(\lambda)}(s) &= \tilde{x}_{2k}^{(\lambda-1)}(s) + \sum_i u_i^{(\lambda)} \cdot \tilde{x}_{2(k-i)+1}(s), \quad \lambda \text{ even} \end{aligned}$$

This means that we are effectively applying the original temporal subband transform directly to the sequence of warped frames, $\tilde{x}_k(s)$. The low-pass temporal subband sequence, $l_k(s) = x_{2k}^{(\Lambda)}(s)$ is then equivalent to the low-pass subband, $\tilde{x}_{2k}^{(\Lambda)}$, of the warped sequence, warped back onto the coordinate system of frame $2k$. Similarly, the high-pass temporal subband sequence, $h_k(s) = x_{2k+1}^{(\Lambda)}(s)$ is obtained by warping $\tilde{x}_{2k+1}^{(\Lambda)}$ back onto the coordinate system of frame $2k+1$.

We have shown that for spatially continuous frames, using an invertible motion model, the proposed motion adaptive lifting steps are equivalent to applying the original subband transform directly to a sequence of warped frames, $\tilde{x}_k(s)$. The warping serves to compensate for the motion trajectories. Equivalently, the original subband transform is being applied along the motion trajectories, $s_k = V_k(s)$.

This suggests a strong connection between the proposed LIMAT framework and the frame warping methods proposed in [4] and [5]. In fact, the key innovation in our proposed method lies

in the way we treat motion fields with locally expansive or contractive behaviour. In direct frame warping methods, the only motion models which allow for perfect reconstruction are those for which the motion-compensating warping operation can be inverted in the discrete domain. Modelling the discrete frames $x_k[\mathbf{n}]$, as unit-spaced samplings of an underlying sequence of continuous frames, $x_k(\mathbf{s})$, the motion warping operation, $\tilde{x}_k(\mathbf{s}) = x_k\left(V_k^{-1}(\mathbf{s})\right)$, can be implemented invertibility in the discrete image domain only if both $\tilde{x}_k(\mathbf{s})$ and $x_k(\mathbf{s})$ are Nyquist band-limited. This means that the functions, V_k, must have Jacobians with determinant no less than 1. Equivalently, the motion field from frame 0 to frame k must not contain any local contractions. For example, a zooming video sequence might be described by $V_k(\mathbf{s}) = \alpha\mathbf{s}$ where $\alpha < 1$. While V_k itself is invertible, the discrete warping of $x_k[\mathbf{n}]$ onto $\tilde{x}_k[\mathbf{n}] = x_k\left(V_k^{-1}(\mathbf{n})\right)$ cannot be inverted, because $\tilde{x}_k[\mathbf{n}]$ uses a reduced number of samples to represent the same spatial region as $x_k[\mathbf{n}]$.

Since the average sample density is the same in each frame of the original video sequence, local expansion in one region of a warped image must be matched by local contraction in another. It follows that the only motion models which can be truly inverted in the discrete domain are those which involve neither local expansion, nor local contraction. Such models are invariably restricted to some combination of translation and skewing.

In the LIMAT framework, the temporal subband transform is not applied directly to the warped frames, $\tilde{x}_k[\mathbf{n}]$. Instead, the update terms in each lifting step are compensated for motion. This ensures that the transform remains invertible, even if the individual frame warping operations are not invertible. In regions where the motion is neither expansive nor contractive, the proposed technique is essentially equivalent to the direct frame warping methods. Even in regions where the motion field is locally expansive or contractive, the low frequency components of each frame can be correctly recovered from its warped counterpart, so that the discrete motion adaptive lifting transform processes these low frequency components in the desired manner. The same is not true for the remaining high frequency components, but images typically have most of their energy concentrated at lower frequencies.

3 Motion Modelling and Preliminary Observations

To realize a subband decomposition along meaningful motion trajectories, the motion in the sequence must be accurately modelled. The LIMAT framework is well suited to this challenging task, since it may incorporate any motion model, without sacrificing the invertibility of the temporal transform. To demonstrate the importance of this property, we use a block-based motion model as a reference, and investigate the improved performance possible with a deformable mesh motion model.

In Section 2.3 we saw that applying the temporal DWT along the scene motion trajectories assumes invertible motion mappings. If the mappings are invertible, only one must be estimated, since the other can be determined by inversion. We adopt this approach in our motion estimation, but note that invertibility only applies to continuous motion fields, such as those yielded by the deformable mesh motion model. Inverting and coding motion fields is discussed further in Section 4. The experimental results presented in this section do not include the cost of coding the motion information, but in Section 5 we see that this has relatively little effect.

3.1 Block Motion Model

Block motion models are predominant in traditional motion-compensated video coders. In a typical block-based motion estimation algorithm, the current frame is partitioned into a regular grid of blocks, and each block undergoes translation to a new location in the reference frame. The motion mapping can be compactly represented by the field of block displacement vectors.

The block model corresponds to a piecewise constant approximation of the underlying motion field. In general, this is only an efficient representation of very smooth motion fields, or those consisting of only simple translational motion. In particular, block motion models poorly represent expansions and contractions in the true motion field, which commonly arise due to deformations of scene objects, camera panning and zooming. This hampers the performance of motion-compensated

video coders for any sequence involving non-trivial motion. Furthermore, the introduction of artificial discontinuities into the motion field can introduce disturbing visual artefacts.

Within the context of LIMAT, an important limitation of discontinuous motion fields is that inverse motion mappings do not exist. Despite this, we incorporate the block motion model into the proposed framework by adopting an ad-hoc procedure for approximating inverse mappings. This approximation involves reversing each motion vector; it is only likely to be accurate in regions where the motion is both smooth and relatively small.

Our experiments use a hierarchical block-matching algorithm. In comparison to full-search approaches, hierarchical motion estimation is more robust to image noise and illumination fluctuations. As we shall see, this is desirable for LIMAT because we are more interested in modelling true motion than simply minimizing the prediction residual. Hierarchical motion estimation also tends to lead to more uniform motion fields, which are more amenable to our approximate inversion strategy. In addition, spatial correlation can be subsequently exploited, leading to efficient lossless coding of the motion overhead.

The reconstructed video PSNR, for a bit rate of 1 Mbps, is given in Table 2. Incorporating motion compensation into the transform generally improves the reconstructed sequence PSNR. Consider, for example, the Mobile Calender sequence, in which increases of 4.94 dB and 5.58 dB are observed for the Haar and 5/3 wavelets, respectively. Similar gains of 4.59 dB and 6.25 dB are made for the Flower Garden sequence, but there is little improvement for the Table Tennis and Football sequences. In these sequences, motion-compensating the Haar lifting steps actually reduces the reconstructed video PSNR.

Although motion adaptation generally increases energy compaction, it can also expand the quantization error energy during synthesis. Most video sequences contain local expansions and contractions, and therefore do not exhibit a one-to-one correspondence between pixel locations in consecutive frames. This is essentially the source of the "disconnected" pixel problem seen in block displacement approaches. It manifests itself in this scheme when displaced blocks overlap in

the reference frame, causing some pixels to be mapped to multiple locations in the current frame. During temporal synthesis, the quantization error in these pixels will also be mapped to multiple locations in the reconstructed frames, causing an overall increase in frame distortion.

Our current quantization and coding strategies treat the 3D transform as though it were fully separable. As a result, PSNR performance is only improved if the increase in energy compaction outweighs quantization error energy expansion during synthesis. The motion adaptive Haar transform does not achieve this with the Football and Table Tennis sequences, because the motion is not sufficiently exploited. The Football sequence contains rapid translations and deformations, along with camera panning and zooming. This is poorly represented by block motion models. Our motion estimation algorithm also struggles to capture the motion of the rapidly translating ball in the Table Tennis sequence. The Mobile Calender and Flower Garden sequences contain more uniform motion, which is more easily captured.

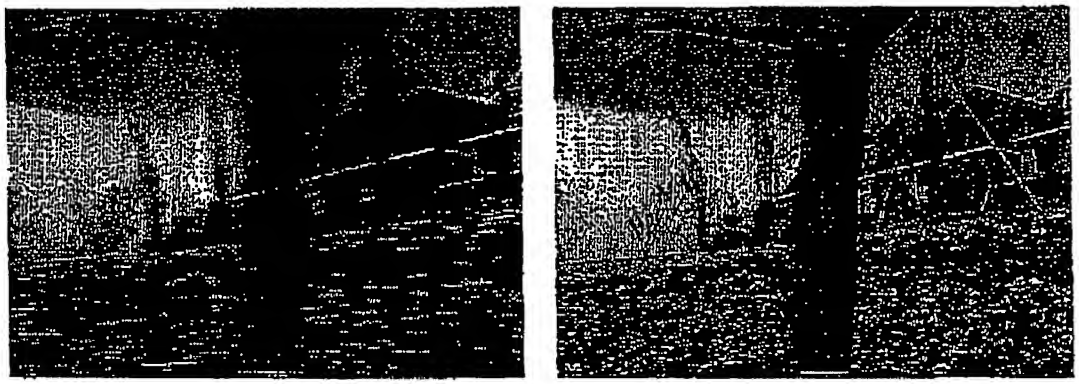
In the case of the 5/3 transform, inadequate motion modelling and estimation is partially alleviated by the bidirectional motion compensation associated with the first lifting step. The corresponding increase in energy compaction leads to improved compression performance in all test sequences. Nevertheless, further improvements should be possible if the expansion in quantization error energy were correctly compensated during quantization of the spatial subband samples.

In all cases, the use of motion-compensated lifting steps significantly improves the visual quality of the low-pass temporal subband. For example, we compare the visual effect of motion compensation on the low-pass frames from the Flower Garden sequence. Fig. 6a shows a low-pass frame produced using three levels of the original 5/3 temporal transform. In Fig. 6b, the substantial ghosting artefacts are avoided completely using motion compensation.

Interestingly, even when motion compensation fails to increase the reconstructed video PSNR, subjective quality may still be improved. A frame from the Table Tennis sequence, reconstructed at 500kbps, is shown in Fig. 7. Here we compare the effect of motion, after 3 levels of Haar temporal transform, on the visual quality of the full frame rate reconstruction. As mentioned, our motion

Table 2: Reconstructed PSNR using block motion model, at 1 Mbps

	No DWT	3 Level Haar			3 Level 5/3		
Sequence		None	Block	Gain	None	Block	Gain
Mobile	19.54	21.40	26.34	+4.94	21.91	27.49	+5.58
Table	28.50	31.59	30.45	−1.14	31.79	33.09	+1.30
Flower	21.11	21.80	26.39	+4.59	21.96	28.21	+6.25
Football	26.25	26.75	25.02	−1.73	26.57	27.30	+0.73



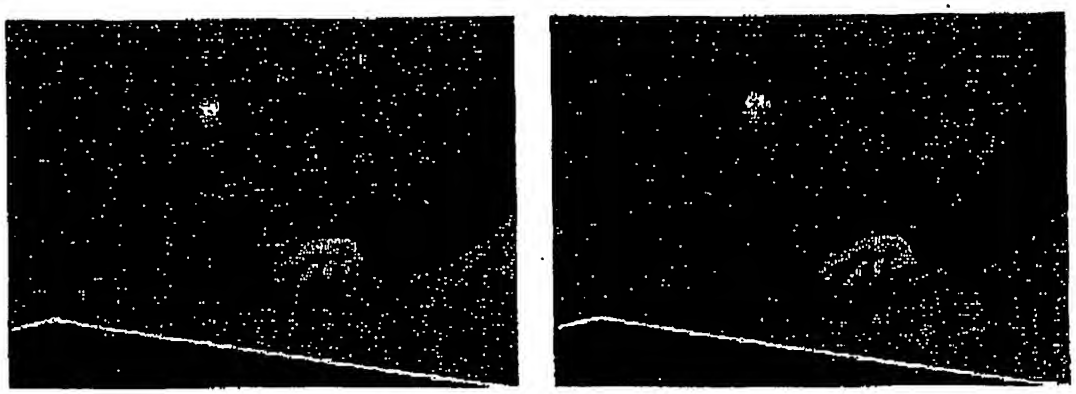
(a) (b)

Figure 6: Demonstrates the visual effect of block-based motion compensation on the low-pass frames, using the 3-level 5/3 temporal DWT. The ghosting artefacts observed in (a) are avoided in (b) by motion compensation.

estimation algorithm does not adequately capture the rapidly translating table tennis ball, resulting in a 0.43 dB reduction in PSNR, but the overall visual quality of the motion-compensated frame is still noticeably better. In particular, observe the well-defined edges of the player's arm, hand and bat. We remark that PSNR is often not a sufficient measure for reconstructed video quality.

3.2 Deformable Mesh Motion Model

Unlike block-based models, deformable meshes can track complex motion, including local expansion and contraction, while maintaining a continuous motion field. The current frame is partitioned into a regular grid of patches, usually quadrilaterals or triangles, whose boundaries form a regular mesh. The mesh node-points move to form a warped mesh on the reference frame, and the mapping is compactly represented by the set of node displacement vectors. The motion vector at any



(a) (b)

Figure 7: Compares reconstructed visual quality of motion-compensated Haar 3-level temporal DWT at 500kbps. Reconstruction PSNR in (a) is 28.22 dB, reduced to 27.78 dB by motion compensation in (b), despite an improvement in visual quality.

given location within a patch is approximated by linearly interpolating the motion vectors at the patch vertices. This corresponds to an affine transformation for triangular meshes, and a bilinear transformation for quadrilateral meshes.

Deformable meshes yield motion fields that are piecewise smooth and continuous at patch boundaries. These motion fields generally provide a much better representation of the underlying motion field than discontinuous block motion fields. A further advantage is the invertibility of continuous motion fields, which lends itself naturally to the LIMAT framework.

Determination of a globally optimum set of node vectors is not usually possible within reasonable computational constraints. Local searches, or gradient-based approximations are commonly used instead, which typically find only a local minimum of an appropriate objective function, such as the energy of the displaced frame difference. Nevertheless, deformable meshes have been found to offer motion compensation superior to block-based models, using the same number of motion parameters [15].

In this work, we incorporate a triangular deformable mesh model into the LIMAT framework, using an estimation algorithm based on the hexagonal refinement strategy proposed in [15]. As

mentioned, local expansions and contractions in the motion field can cause an increase in reconstruction error energy during synthesis. With a deformable mesh, the expansion in quantization error energy is directly related to expansion in the mesh itself, which is given by the determinant of the affine transform associated with each triangular patch in the mesh. To discourage unnecessary error expansion, we weight the distortion within each patch by the determinant of its affine transform. This leads to a significant improvement in reconstruction PSNR. In future work, further improvement could be obtained by directly adapting the spatial quantization and coding of the temporal subbands, according to the transform determinants.

Table 3 shows the reconstruction PSNR using the mesh motion model, with three levels of temporal transform, and a bit rate of 1 Mbps. Unlike the block-based motion model, mesh-based motion compensation of the lifting steps improves the compression performance in every sequence. The mesh model also consistently outperforms the block motion model, although the improvement is marginal in the case of the Table Tennis sequence, with the 5/3 transform. This is because block-based motion models are better suited to representing the highly discontinuous motion field in the neighbourhood of the rapidly translating table tennis ball.

We have shown that the use of a superior motion model leads to improved video compression within the LIMAT framework. Of course, a variety of more sophisticated motion modelling techniques exist in the literature. For example, it is known that spatially non-uniform motion models can achieve superior motion compensation by allocating a denser distribution of motion vectors to regions with complex motion. We have also observed that our mesh motion estimation is less effective for larger temporal displacements. This suggests a hybrid approach may be useful, possibly employing block-based or global motion models for higher levels in the temporal subband decomposition.

Table 3: Reconstructed PSNR using deformable mesh motion model, at 1 Mbps

	3 Level Haar				3 level 5/3			
Sequence	None	Mesh	Gain	c.f. Block	None	Mesh	Gain	c.f. Block
Mobile	21.40	26.69	+5.29	+0.35	21.91	27.73	+5.82	+0.24
Table	31.59	32.00	+0.41	+1.55	31.79	33.16	+1.37	+0.07
Flower	21.80	27.49	+5.69	+1.10	21.96	28.91	+6.95	+0.70
Football	26.75	27.51	+0.76	+2.49	26.57	28.19	+1.62	+0.89

3.3 The Importance Of Establishing Motion Trajectories

In Section 2.3, we showed that motion-compensating the lifting steps is equivalent to applying the temporal DWT along an underlying set of motion trajectories, so long as the motion model is invertible. Accordingly, our results so far have involved motion mappings, $\mathcal{W}_{k_2 \to k_1}$ and $\mathcal{W}_{k_1 \to k_2}$, which are related through the requirement that $\mathcal{W}_{k_2 \to k_1}$ should approximately invert $\mathcal{W}_{k_1 \to k_2}$. In particular, we estimate only one set of motion parameters for each such pair of mappings.

It is tempting to independently optimize the parameters of each individual motion mapping, with respect to a displaced frame difference measure. In this case, $\mathcal{W}_{k_1 \to k_2}$ and $\mathcal{W}_{k_2 \to k_1}$ are obtained through independent forward and backward motion estimation. The resulting motion maps will not generally be inverses of one another. Even in the absence of modelling or estimation errors, discrepancies between $\mathcal{W}_{k_1 \to k_2}$ and $\mathcal{W}^{-1}_{k_2 \to k_1}$ (if it exists) can be expected in regions of occlusion and uncovered background. In such regions the relationship between successive frames cannot truly be described in terms of a set of motion trajectories. Nevertheless, if we choose to use independently optimized motion mappings, abandoning motion trajectories, we find in practice that compression performance suffers significantly.

To quantify this effect, we compare the reconstructed PSNR obtained using directly inverted motion maps, with that obtained by estimating every motion map independently. The results are taken using 3 levels of temporal transform, at a compressed bit rate of 1 Mbps. According to Table 4, the reconstructed PSNR is uniformly higher with inverted motion fields, by as much as 2.91 dB in one case. Note that the improvement is also consistently larger for the deformable mesh model, as compared with the block motion model, which lacks a true inverse. These results reinforce our

Table 4: Gain in reconstruction PSNR from using inverted motion fields instead of estimated motion fields, at 1 Mbps

Sequence	Haar, Block	Haar, Mesh	5/3, Block	5/3, Mesh
Mobile	+0.21	+0.29	+0.10	+0.14
Table	+0.42	+2.12	+0.13	+0.67
Flower	+0.14	+0.56	+0.05	+0.31
Football	+0.68	+2.91	+0.31	+1.22

earlier analysis, which provides a meaningful interpretation to the LIMAT framework only in the context of an invertible motion model. Even if the motion trajectories assigned by the model do not perfectly describe the underlying scene, at least we know that the wavelet filters are being applied along those trajectories. When $\mathcal{W}_{k_1 \to k_2}$ and $\mathcal{W}_{k_2 \to k_1}$ are not inverses of one another, there is no clear way to understand the behaviour of the transform, but our experimental observations suggest that the existence of motion trajectories is important for compression performance.

4 Motion Representation

Each level of the motion adaptive lifting transform requires a distinct motion mapping for every lifting filter tap. For example, the Haar wavelet requires one mapping per frame; repeated at each level of decomposition. With multiple levels this is approximately two mappings per original frame. The 5/3 transform requires double this number of mappings. The cost of coding this motion information, and the computational effort involved in estimating it can be quite considerable. However, strong dependencies exist between the various motion mappings. We exploit these dependencies in our motion representation to reduce the number of distinct mappings to a maximum of one per original frame, regardless of the wavelet kernel. The number of motion mappings is halved for the Haar wavelet and reduced to a quarter for the 5/3 wavelet. As a result, the motion overhead is comparable to that required by motion-compensated predictive coders.

In Section 3 we saw that by using forward and backward motion mappings that are inverses, the temporal transform is being applied along motion trajectories. Apart from yielding superior

compression performance, this has the added benefit that only one set of parameters is required for each forward-backward pair. We also exploit the fact that motion mappings at higher subband levels can be considered as essentially a concatenation of two consecutive motion mappings from the previous level.

4.1 Examples with the Haar and 5/3 Transforms

The motion representation for the 5/3 wavelet, using two transform levels, is given in Fig. 8. The mappings required to perform the lifting steps are shown as arrows, where the i^{th} forward mapping in the j^{th} transform level is denoted $\mathcal{F}_i^j$. We find it intuitive to define forward mappings as those which approximate the current frame by warping a frame with a lower time index. Likewise, backward mappings, denoted $\mathcal{B}_i^j$, correspond to warping a frame with a greater time index, to spatially align with the current frame. Observe that the entire set of motion fields can be represented by only $\mathcal{F}_1^2$ and $\mathcal{B}_2^1$. Inverting $\mathcal{F}_1^2$ produces the backward mapping $\mathcal{B}_1^2$. The forward mapping $\mathcal{F}_1^1$ is recovered by concatenating the upper-level forward mapping $\mathcal{F}_1^2$ with the lower-level backward mapping $\mathcal{B}_2^1$. The remaining mappings $\mathcal{B}_1^1$ and $\mathcal{F}_2^1$ are recovered by inverting $\mathcal{F}_1^1$ and $\mathcal{B}_2^1$, respectively. The case for the Haar wavelet is a simplification of the above. Mappings $\mathcal{F}_2^1$ and $\mathcal{B}_2^1$ are not required, so we simply code mappings $\mathcal{F}_1^1$ and $\mathcal{F}_1^2$, and recover the corresponding backward motion fields by inversion. This procedure can be iterated to any number of transform levels, and the total number of required mappings is upper bound by one per original frame. Note that inverting any level of the transform requires no motion information from lower levels, so temporal scalability is not sacrificed.

Evidently, a motion mapping between any pair of frames can be obtained by a combination of concatenations and inversions involving the sequence of mappings $\mathcal{F}_i^2$ and $\mathcal{B}_{2i}^1$. It follows that this motion representation is sufficient for any wavelet kernel.

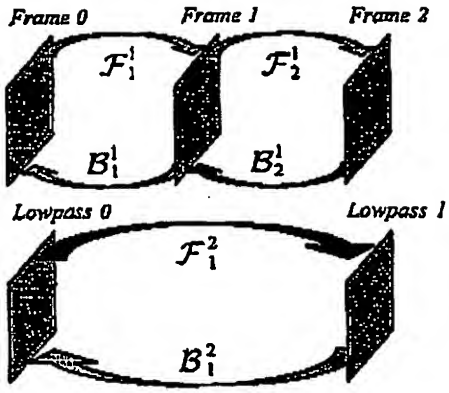


Figure 8: Motion representation for two levels of 5/3 temporal DWT. The grey mappings are inferred from the coded mappings, shown in black.

4.2 Inverting and Coding Motion Mappings

Our motion representation depends on our ability to compute the inverse of a motion field. To be invertible, mappings must be continuous and one-to-one. This is not the case for the block motion model, so we adopt an ad-hoc approach that simply involves reversing each original motion vector. This is a good approximation when the motion is smooth and relatively small, but as observed in Section 3.3, compression performance of the block-based model is inevitably limited by the lack of a true inverse.

To invert the deformable mesh motion field, the affine transformation corresponding to every pixel location is determined and inverted. This process cannot be represented by the warping of another regular mesh in the reverse direction. However, such a representation is unnecessary, since the true inverse can be applied to each pixel during motion compensation. We note that this leads to slightly higher complexity compared with normal motion compensation, since finding the patch belonging to each pixel is more difficult in a warped mesh than in a regular mesh. We also note that the mesh is non-invertible in any region where the mesh folds over itself, because the one-to-one nature of the mapping is lost. We avoid this in our estimation algorithm by disallowing a negative transformation determinant. Accurate motion modelling requires the mesh to detach from frame

boundaries. However, wherever the mesh moves away from the boundary towards the centre of the image, some pixels are inevitably left without a corresponding inverse. In our algorithm, the unknown inverse transformations are approximated by nearest-neighbour extrapolation from the deformed mesh, in an attempt to maintain continuity of the motion field. We use a zero-order hold boundary extension policy to allow the mesh to pass over the boundaries without introducing discontinuities in the motion field.

The concatenation of multiple motion mappings cannot be represented by a single normal motion mapping. Instead, each pixel should be warped once for every individual mapping that affects it. While possible, this would become very difficult for multiple concatenations, as occurs with several transform levels. Moreover, the concatenation of multiple motion fields does not necessarily produce the most effective combined motion field because inadequacies in the individual motion fields tend to propagate.

To rectify this problem in the case of the block motion model, we approximate a concatenated mapping based on each block's trajectory through the two motion fields. Similarly, with the mesh motion model, we approximate the concatenation by tracking the location of mesh node-points through the two mappings, resulting in a new mesh.

To avoid possible loss in performance due to our approximate representation of concatenated mappings, our coder may opt to encode refinement fields for any mapping inferred via concatenation. A refinement field is the difference between an inferred field and that obtained by direct estimation. Note that refinement fields are irrelevant for the Haar transform, which involves no concatenated mappings. The 5/3 may require up to one refinement field for every pair of mappings, at each level. Longer wavelet kernels may require more. In any event, the refinement fields generally consist of small values, mostly zeros, which are less costly to encode.

This brings us to the classic problem of optimal bit rate distribution between the motion information and the subband coefficients. For the purpose of these experiments, we circumvent this issue by coding all refinement fields. At low bit rates the motion cost becomes more significant and

superior results may be achieved by selectively coding the refinement fields.

To code the motion information, we first spatially predict the directly estimated fields, then losslessly code the residual fields, along with the refinement fields, using category codes [13]. Category codes are useful here because we expect the likelihood of the refinement vectors to be inversely proportional to their magnitude. We model an arithmetic coder to determine the cost of coding the category symbols.

It is worth noting that this motion representation strategy is well aligned with efficient methods for motion estimation. Refinement fields are estimated using the inferred field as an initial guess, resulting in much less computation. In total, only one full motion estimation operation is required per original frame, which is comparable with simple motion-compensated prediction methods.

5 Experimental Results

We provide results for the block-based motion model and the triangular deformable mesh model, using three levels of temporal transform. We use the first 96 frames of the standard test sequences, Mobile Calender, Table Tennis, Flower Garden and Football. The original full colour sequences have a frame rate of 30 fps and a spatial resolution of 352×240. Chrominance components are subsampled by 2 in both spatial dimensions.

The block-based model is implemented using a hierarchical search method. A coarse motion field is first estimated by full-search block matching at half the spatial resolution. The search range is relative to the temporal displacement of +/-8 pixels per frame, except for the Football sequence where a search range of +/-16 is used. The coarse motion field is successively refined on interpolated frames up to eighth-pixel accuracy. The block size is 16×16, giving 330 motion vectors per field. We estimate the motion fields for all transform levels using original frames, with a mean-squared error distortion measure.

We form the regular triangular mesh by dividing 16×16 blocks along their diagonals. Motion vectors are estimated for each node in the mesh, resulting in 368 vectors per motion mapping.

A coarse motion field is estimated by full-search block matching at half the spatial resolution, with overlapping blocks centred at each node-point. We use the same search range as with the block-based motion estimation algorithm. The motion field is refined using the hexagonal refinement algorithm proposed in [15], repeated at successively higher resolutions until the motion field is eighth-pixel accurate. As with the block model, the motion estimates are based on original frames, but the mean-squared error distortion measure is now weighted by the determinant of the corresponding affine transformation. As discussed in Section 3.2, this is to discourage excessive expansion in the mesh.

The temporal subband frames are subjected to spatial wavelet decomposition and embedded block coding of the quantized wavelet coefficients, using an implementation of the JPEG2000 image compression standard. Although results are given only for luminance components, each colour component is assigned equal importance during rate allocation. A constant number of bits are allocated to each group of 8 original frames.

Tables 5 and 6 give reconstruction PSNR at bit rates of 1 Mbps and 500 kbps, respectively. This includes the cost of coding the motion information. To emphasize the scalability of the compression system, the test bit rates were obtained by simply discarding unwanted bits from another bit-stream compressed to a much higher bit rate.

Including the cost of coding the motion information does not affect the conclusions drawn so far. Firstly, the 5/3 wavelet still consistently outperforms the Haar transform. The improvement is slightly diminished for the 500 kbps case, where the added cost of coding the refinement fields is more significant than at 1 Mbps. Selective coding of the refinement fields is likely to have been beneficial here. Secondly, the mesh motion model is still largely superior to the block model, even though the more spatially coherent block motion fields are less costly to encode.

As a reference, we compare the LIMAT framework with the motion-compensated 3D Subband Coder (MC-3DSBC) proposed in [7]. As with all block displacement approaches, the temporal transform in MC-3dSBC is restricted to the Haar wavelet and block-based motion compensation.

Table 5: PSNR performance of LIMAT at 1 Mbps

Sequence	No DWT	Haar	Haar, Block	Haar, Mesh	5/3	5/3, Block	5/3, Mesh
Mobile	19.54	+1.86	+6.70	+7.01	+2.37	+7.49	+7.59
Table	28.50	+3.09	+1.59	+3.16	+3.29	+4.09	+4.12
Flower	21.11	+0.69	+4.60	+5.86	+0.85	+6.48	+7.01
Football	26.25	+0.50	−1.79	+0.72	+0.32	+0.23	+1.05

Table 6: PSNR performance of LIMAT at 500 kbps

Sequence	No DWT	Haar	Haar, Block	Haar, Mesh	5/3	5/3, Block	5/3, Mesh
Mobile	17.98	+1.75	+5.23	+5.49	+2.05	+5.79	+5.68
Table	26.05	+2.81	+1.39	+2.86	+3.16	+3.71	+3.72
Flower	18.96	+0.78	+4.13	+4.82	+0.86	+5.65	+5.73
Football	23.90	+0.39	−1.95	+0.62	+0.31	−0.03	+0.64

In Table 7 we compare our results with those in [7], at a bit rate of 1.2Mbps.

We first compare the performance of the two coders in the context of block-based motion modelling with the Haar transform. In comparison with the block motion model presented in this work, the authors in [7] have used a more sophisticated hierarchical model incorporating variable-sized blocks. As mentioned, our block-based motion estimation struggles to capture fast-moving objects, as seen clearly in Fig. 7b. As a consequence, our implementation yields inferior results in comparison with the reference coder, particularly for the test sequences that contain rapid motion.

Of course, the proposed scheme is much more effective with the 5/3 wavelet and deformable mesh motion modelling, despite the shortcomings of our motion estimation algorithms. In fact, our implementation outperforms MC-3DSBC for all sequences excluding Table Tennis. We note that the highly inhomogenous motion field in Table Tennis can be represented much more effectively by spatially adaptive motion models, such as that used by the reference coder.

Despite these highly competitive results, it is important to realise that the purpose of these experiments is not to evaluate the performance of a fully developed video coding system, but to justify a new framework in which superior video coders may be built.

Table 7: PSNR performance of LIMAT, compared with MC-3DSBC, at 1.2 Mbps

Sequence	MC-3DSBC	Haar, Block	Haar, Mesh	5/3, Block	5/3, Mesh
Mobile	27.01	−0.12	+0.22	+0.22	+1.04
Table	33.78	−2.74	−1.27	−1.27	−0.47
Flower	27.55	−0.53	+0.34	+1.24	+1.77
Football	28.02	−2.63	+0.07	−0.62	+0.18

6 Conclusions

We propose a new framework for the construction of invertible motion adaptive temporal transforms. The LIMAT framework is based on the lifting representation of the temporal DWT, with motion-compensated lifting steps. Invertibility is an inherent property of the lifting structure, irrespective of the manner in which we model or compensate for motion.

Incorporation of sophisticated motion models allows the transform to adapt to complex motion. We demonstrate this with a deformable mesh motion model. Deformable meshes can improve motion compensation by tracking expansions and contractions, while maintaining a continuous motion field. More importantly, only continuous motion mappings allow us to understand the proposed transform as truly performing a subband decomposition along the motion trajectories. Our experimental results show that this is particularly desirable for compression performance.

By contrast to block-based and frame-warping approaches, the LIMAT framework is amenable to any temporal wavelet kernel. In fact, we observe superior performance with the 5/3 wavelet as compared to the Haar transform. This is explained by improved energy compaction along the motion trajectories, also because bidirectional motion compensation in the first lifting step helps reduce the effects of inaccuracy in the motion model.

We provide a compact representation for the motion parameters, exploiting the redundancy between the distinct motion fields used in the transform. The resulting motion overhead is comparable to that of motion-compensated prediction coders.

Future research will focus on incorporating more sophisticated invertible motion models, and adapting spatial quantization in regions of expansion and contraction.

References

[1] G. Karlsson and M. Vetterli, "Three-dimensional subband coding of video," *Proc. Int. Conf. Acoust. Speech and Sig. Proc.*, vol. 2, pp. 1100–1103, Apr 1988.

[2] B.J. Kim and W.A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," *Proc. IEEE Data Compression Conf. (Snowbird)*, pp. 251–260, Mar 1997.

[3] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circ. Syst. for Video Tech.*, pp. 243–250, June 1996.

[4] D.S. Taubman and A. Zakhor, "Multi-rate 3-d subband coding of video," *IEEE Trans. Image Proc.*, vol. 3, no. 5, pp. 572–588, September 1994.

[5] J. Tham, S. Ranganath, and A. Kassim, "Highly scalable wavelet-based video codec for very low bit-rate environment," *IEEE Journal on Selected Areas in Comm.*, vol. 16, pp. 12–27, Jan 1998.

[6] J. Ohm, "Three dimensional subband coding with motion compensation," *IEEE Trans. Image Proc.*, vol. 3, pp. 559–571, Sep 1994.

[7] S. Choi and J. Woods, "Motion compensated 3-D subband coding of video," *IEEE Trans. Image Proc.*, vol. 8, pp. 155–167, Feb 1999.

[8] A. Secker and D. Taubman, "Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting," *Proc. IEEE Int. Conf. Image Proc.*, pp. 1029–1032, Oct 2001.

[9] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Applied and Computational Harmonic Analysis*, vol. 3, no. 2, pp. 186–200, April 1996.

[10] R. Calderbank, I. Daubechies, W. Sweldens, and B. Yeo, "Wavelet transforms that map integers to integers," *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332–369, July 1998.

[11] V.K. Heer and H.-E. Reinfelder, "A comparison of reversible methods for data compression," *Proc. SPIE conference, 'Medical Imaging IV'*, vol. 1233, pp. 354–365, 1990.

[12] D. Le Gall and A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 761–764, April 1988.

[13] D.S. Taubman and M.W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, Boston, 2002.

[14] J. Ohm, "Advanced packet-video coding based on layered VQ and SBC techniques," *IEEE Trans. Circ. Syst. for Video Tech.*, vol. 3, no. 3, pp. 208–221, June 1993.

[15] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Trans. Circ. Syst. for Video Tech.*, vol. 4, pp. 339–367, Jun 1994.